Cost-Efficient Detection of Adverse Drug Reaction in User Generated Content

Anonymous Author(s)

ABSTRACT

Discovering adverse drug reactions (ADRs) is a critical component of drug safety. In addition to controlled clinical trials, continuous monitoring of drug effects after market introduction provides valuable insights into ADRs. A premier source of monitoring ADRs in the wild is social media where users share and discuss their personal experiences with drugs using informal laymen terminology. Existing methods for automatic ADR detection require extensive human supervision and training, thus limiting their effectiveness and adaptability to the high dynamicity of layman terminology.

This paper introduces a novel supervised approach for ADR detection from social media posts. We specifically address the problem of cost for supervised training by using deep probabilistic variational autoencoders to automatically generating large training datasets from a small number 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 has comparable performance to fully supervised techniques while drastically lowering the demand for labeled training data, allowing us to maintain performance with down to only 25% of training data.

CCS CONCEPTS

• **Information systems → Data extraction and integration**; • **Applied computing → Consumer health**; • **Computing methodologies** → *Semi-supervised learning settings*;

KEYWORDS

Web Information Extraction, Adverse Drug Reaction Detection, Training Data Generation, Deep Language Generative Models

ACM Reference Format:

Anonymous Author(s). 2019. Cost-Efficient Detection of Adverse Drug Reaction in User Generated Content. In *Proceedings of The Web Conference (WWW'19).* ACM, New York, NY, USA, Article 4, 11 pages. https://doi.org/10.475/123_4

1 INTRODUCTION

Monitoring the effects of medical drugs after they have been released into the market is an important element of drug safety. Adverse Drug Reactions (ADRs) – the harmful reactions caused by taking a medication – is the fourth leading cause of death in the United States [9]. Studies have shown that clinical trials are not able to fully characterize drugs' adverse effects [1, 9, 16]. 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 [42]. Nowadays, the Web is becoming a preferred channel for millions of users and patients to share, discuss, and seek health information [18] in Social Media. This user generated content can provide valuable insights for monitoring public health [2, 38], and especially provides new chances for monitoring adverse drug reactions from an additional point of view [30, 43].

However, web users report ADRs using a different language style and terminology that depends on the user's medical proficiency, but also on the type of online medium (e.g., health forums vs micropost social networks). Therefore, ADR reports from user generated content typically differ significantly from ADR statements in professional medical text. As exemplified in Table 1, laymen often use diverse dialects [22] when describe medical concepts, and make abundant use of figures of speech (e.g. metaphors) and informal terminology. Additionally, social media text is usually informal and succinct, often due to limitations imposed by the communication platform, limiting thus the extent and semantic richness of the report [3].

Table 1: Examples of ADR (in bold) from different sources.

Professional	Patient complains about **Insomnia**. It Started 3 days ago
AskAPatient	I took evista for the first time about 15 years ago. It was the worst year of my life. **No sleep** and constant **night sweats**
Twitter	Exhausted...**can't fall asleep**. Don't wanna take a trazadone and wake up hungover. #Sleepdisorderproblems

Approaches for detecting terms in informal medical language – like ADRs – often rely on (semi-)manually generated dictionaries (e.g. laymen health dictionaries), or supervised machine-learning-based sequence classifiers. Due to the language dynamicity in online and offline communication [24, 51], 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 [12]. While existing methods such as distant supervision [35] or bootstrapping techniques [47] can reduce the cost of training data creation, they are not suitable for detecting ADRs that are scarcely addressed in existing dictionaries and are diversely expressed in different sources. More recently, researchers have started to investigate techniques for expanding the size of manually created training data. Often, sentence similarity

WWW'19, May 2019, San Fransisco USA

ACM ISBN 123-4567-24-567/08/06.
https://doi.org/10.475/123_4

implemented with embeddings [27, 34] is used to discover similar sentences, and then annotations are automatically propagated to those sentences [32]. While these techniques have indeed shown to reduce the cost of training, we will argue in this paper that they are typically limited by sentence availability as the reliability of annotation propagation suffers when sentences are not similar enough.

Therefore, we focus on the following research question:

> **RQ**: 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 carefully generate artificial sentences closely mimicking existing training data which are then annotated automatically via label propagation to minimize training costs for ADR recognition in user generated content. This contrasts existing approaches expanding the manually created training data set by discovering additional sentences in an existing dataset. While this is not part of our evaluation, we speculate that it can be generalized to any supervised text-sequence classification problem (like Named Entity Recognition).

Our approach covers four steps: first we learn sentence distributions from an existing dataset with a small human-annotated subset (e.g., the complete text of a large health forum with a very small subset annotated with ADRs labels), and generate new artificial data samples resembling the labeled ones. Then, we propagate labels to these new sentences by relying on already known labels. These new labeled sentences are then used for training an ADR detector with much higher quality compared to only relying on the human-annotated training set. This allows us to reduce the annotation costs while maintaining ADR detection quality, as our approach works particularly well with small annotated sets.

To learn high-quality data distributions for artificial text generation, we build our method upon variational autoencoders (VAE), a deep probabilistic neural model which captures latent text features very effectively. Owing to the non-linearity and multi-layer structure of the neural model, the used deep neural networks are better at capturing fine-grained semantics than commonly used shallow networks employed in e.g., word embeddings. Deep probabilistic models thus benefit from both the flexibility of neural networks in learning the underlying data structure and the expressiveness of probabilistic modeling in capturing data distributions. 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 set, while still taking advantage of the implicit semantics contained in the larger dataset, thus effectively increasing the size of training data.

We compare our approach to state-of-the-art ADR detection algorithms on Twitter (using a standard dataset for ADR detection), and on a large new dataset for the Reddit platform we created with the help of expert annotators, and which we make available to the community.

Our experiments with two different sequence-labeling models (CRF and BLSTM-RNN) shows that our approach can achieve superior or comparable performance using significantly less training data (reduced by 75%) than other training methods. For example, when employing an CRF model, only 173 training samples are needed to achieve a better F1-score of 0.53 than using 693 samples without our approach (F1-score of 0.51) on a corpus of 146k Tweets. We also compare to other established techniques like dictionary-based method (QuickUMLS) and alternative training expansion techniques (Self Training and Doc2vec) which are all outperformed by our approach. Finally, to widen the scope of our evaluation, we perform a qualitative analysis to investigate the properties and nature of the prediction mistakes of our approach.

2 RELATED WORK

In this section, we first discuss relevant work about ADR detection in user generated content, and then review relevant automatic training data expansion techniques and generative models which serve as the foundation of our work.

2.1 ADR Detection in User Generated Content

Social Web systems such as forums, blogs, and social media are a popular place for patients to share their health experiences. By analysing the content produced in these systems, it is possible to access valuable information on drug side effects directly from the patients. There is a large body of work on ADR detection in social media such as *Twitter* [10, 37], or forums like *Dailystrength*[28, 36], and *askApatient*. Existing methods for automatic detection of ADRs from in UGC fall into four categories: 1) *dictionary matching approaches*[28, 45], which employ huge dictionaries of medical concepts (e.g. Unified Medical Language System UMLS[1]) that also contain non-professional health vocabulary; 2) *co-occurrence analysis approaches* [18, 20], which find new expressions by expanding dictionary terms with those that frequently co-occur; 3) *Pattern mining techniques* [36, 50] which extract the underlying patterns of user reviews mentioning ADRs; and 4) *supervised machine-learning approaches* [6, 10, 29], which rely on human-labeled training data.

Dictionary-based and co-occurrence analysis approaches are limited by their applicability due to the lack of ADR terminology that is common in UGC, and are expensive to maintain. Supervised and pattern mining approaches rely on ADR annotated training data [19], which is expensive to create as it requires medical expertise for content annotation. Furthermore, the training data is difficult to maintain as it requires to be updated over time along with the language evolution. In our paper, we address this problem by artificially generating training data, thus reducing annotation costs considerably. Moreover, due to the flexibility of the generation model and low annotation costs, this allows us also to address the challenge of language evolution.

Some recent work has started to address the issues of size and cost of ADR training data [11, 30, 37]. Lee [30] 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

[1] https://www.nlm.nih.gov/research/umls/

posts rather than just classifying the whole post as containing an ADR mention. Nikfarjam [37] and Cocos [11] 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 [11] technique, which achieves state-of-the-art performance, is also evaluated in Section 4.1. 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. Therefore, it can be combined with approaches like BLSTM-RNN for even greater effect.

2.2 Automatic Training Data Generation

The availability and the high cost associated with annotating training data has become a key bottleneck in training machine learning models [40]. In recent years, many attempts have been made to reduce annotation costs. In active learning, for instance, users are asked to annotate only the fraction of training data that is expected to be most effective for model training [15]. Entity Set Expansion, on the other hand, is a technique for automatic training data expansion [32] that aims at finding similar entities to a given small set of seed entities [23, 53].

A recent advance that falls between human supervision and fully automatic data expansion method is weak supervision, where unlabeled data is semi-automatically labeled using distant supervision [35], bootstrapping [47], or embedding based methods [27, 34]. Distant supervision relies on external knowledge bases for generating training data. Such knowledge bases are, however, only available for some domains and are rather incomplete for ADR detection. Bootstrapping [47] uses a set of seed terms and extracts features such as unigrams, bigrams, left unigram, and closest verb, to extract new entity mentions in an iterative fashion until no new features are detected. A typical example is self-training [5], which trains a classifier on a small set of labeled data and uses it to increase the amount of annotated data by iteratively applying it to label additional sentences from a corpus and use them to re-train the classifier. These techniques, however, are limited by the learning capabilities of the classifier in capturing both the feature-class relationship and the complex dependencies among words, which are critical for data expansion.

Different from the above approaches, embedding based methods [27, 34] 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, suffer from two major limitations. First, these methods are often limited in capturing fine-grained semantics due to the limited expressiveness of shallow networks adopted by embedding models. More importantly, they 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. We leverage deep probabilistic modeling to learn data distributions from the corpus while capturing the underlying data structure. By doing so, our approach can not only generate new sentences that are semantically meaningful but also reliably label them.

2.3 Variational Autoencoder On Textual Data

Our proposed method is built upon variational autoencoder (VAE) [25, 41], a representative deep probabilistic model that is capable of learning data distributions in the latent feature space that captures the underlying data structure. VAE has been mainly studied for generating image data [13, 39] and has recently been explored in other applications such as information retrieval [8] and recommender systems [31].

Due to the importance of modeling complex semantic structures of the sentences, some recent work has explored VAE for language problems [33, 46, 52]. Miao et al. [33] applied VAE to question answer selection problem (i.e. identifying the correct answer to a question from a set of candidate sentences) and bag of-words representations. Zhang et al. [52] applied VAE for machine translation from German to English and vice versa. Srivastava et al. [46] employed neural variational inference to train topic models.

Our approach for generating additional labeled training data is inspired by [7], where VAEs are used to learn a generative model of text for sentence generation. This work (i.e. [7]), 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.

3 ADVERSE DRUG REACTION DETECTION IN USER GENERATED CONTENT

This section introduces our proposed approach for automatically generating new labeled sentences to improve the training of an ADR detector for a given source of User Generated Content (UGC). In the following, we first give an overview of the approach, and then introduce in more detail its main steps.

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 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 *LP*, 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 underlying structure 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 *SP* 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 distribution from a large text corpus, which can then be used to generate

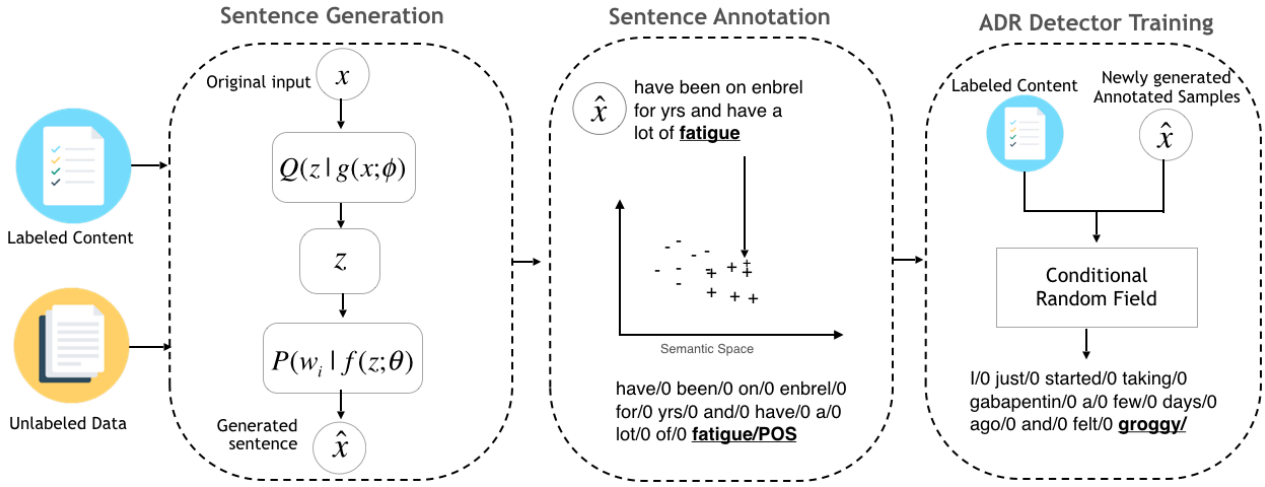

Figure 1: Overview of the proposed cost-efficient Adverse Drug Reaction Detection Approach

posts *SC* semantically similar to a given set of existing labeled content *LC*. Considering for instance the Twitter dataset in our experiments (Section 4.1), the approach allowed us to expand the training set from the 643 tweets to 7,073 tweets. Notice that the method can generate an unlimited number of new sentences, thus extensively cover new variants of existing sentences to better train the ADR slang term detector.

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 sequence 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 [25, 33], which was originally proposed to generate data instances similar to those already in a given dataset. Here, data is embedded (encoded) in 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 [34] 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 2 shows 3 artificial samples generated for 3 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 function $P_\theta(\hat{\mathbf{x}}|\mathbf{z})$ 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:

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

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:

$$\mathcal{L} = \mathbb{E}_{Q(\mathbf{z}|g(\mathbf{x};\phi)}[\sum_{i=1}^{|\mathbf{x}|} \log P(\mathbf{w}_i|f(\mathbf{z};\theta)] - D_{KL}[Q(\mathbf{z}|g(\mathbf{x};\phi)\|P(\mathbf{z})] \quad (2)$$

This is generally known as the evidence lower bound (ELBO) [4]. 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 $g(\mathbf{x};\phi)$.

Table 2: 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

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:

$$P(\hat{\mathbf{x}}|\mathbf{x}) = \int \sum_{i=1}^{|\hat{\mathbf{x}}|} P(\mathbf{w}_i|f(\mathbf{z};\theta))\, Q(\mathbf{z}|g(\mathbf{x};\phi))d\mathbf{z} \tag{3}$$

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 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 training data, thus improving the recall of the ADR detector. In health-related user generated content (i.e. expressing the reactions of taking an drug) it is common for similar adverse drug reactions to be in close proximity, e.g. "After taking Macrobid I got body chills, fever, no appetite, weakness". This step is therefore designed to test and exploit this hypothesis.

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 [17]), while `wheelchair` shall be added to $NEGTerms$. To this end, we use the popular *word2vec* implementation of skip-n-gram word embeddings [34]. 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:

$$SR_{pos}(ut_i) = \frac{\sum_{pterm \in POSTerms} SR_{pos}(ut_i, pterm)}{|POSTerms|} \tag{4}$$

$$SR_{neg}(ut_i) = \frac{\sum_{nterm \in NEGTerms} SR_{neg}(ut_i, nterm)}{|NEgTerms|} \tag{5}$$

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)*. We automatically annotate the terms as follows:

$$ANN(ut_i) = \begin{cases} POS & ut_i \in POSTerms \vee \\ & (SR_{pos}(ut_i) > SR_{neg}(ut_i) \wedge \\ & |SR_{pos}(ut_i) - SR_{neg}(ut_i)| > \text{th}\ ) \\ O & \text{Otherwise} \end{cases}$$

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 [26]; the goal of CRF is to learn the hidden structure of an input sequence. This is done by defining a set of feature functions (e.g. word features, current position of the word labels of the nearby word), assigning them weights and transforming them to a probability to detect the output label of a given entity. We used the popular Conditional Random Field (CRF) sequence model[2] trained using the features listed in Table 3. Finally, the trained ADR detector can be used to detect the ADR mentions in our desired user generated content.

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

Table 3: CRF training parameters.

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

4 EVALUATION

In this section, we introduce the setup of our quantitative evaluation, which develops in three experiments:

Comparison with other ADR Detectors: We compare the performance of our training approach to other ADR techniques, including dictionary-based approaches, and the supervised approaches Conditional Random Fields (CRF) and BLSTM-RNNs. Our approach can be used to train both CRF and BLSTM-RNN.

Effects of Training Data Size and Sampling Numbers: In this experiment, we focus on different setups of our techniques, with increasingly smaller training datasets. We study two independent variables: the size of the human-labeled training dataset, and the number of artificial samples generated per existing annotated sentence. We can show that the effectiveness of our approach increases the smaller training data is, and that generating more samples is typically more effective.

Comparison of Data Expansion Techniques: Our approach expands the size of the human-annotated training data set with additional artificial training data. In this experiment, we compare to Self-Training and Doc2Vec-based techniques to training data expansion, which rely on expanding with existing sentences from an unlabeled dataset. We can show that our approach consistently achieves better performance, with higher precision and comparable recall to the established Doc2Vec expansion technique.

4.1 Experimental Settings

We evaluate the performance of our approach using precision, recall and f-score via approximate matching[48]. In approximate matching a predicted ADR which is a substring of the human-annotated (ADR or vice versa) is considered as true positive. We compare the performance of our approach against established ADR detectors (i.e. dictionary based and fully supervised ADR detectors) and data expansion techniques (i.e. self-training and document embeddings). 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. To provide more insights on the observed performance, we perform a manual qualitative analysis to investigate the properties and nature of the prediction mistakes of our approach.

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 [3]. 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. This extensive dataset will be made publicly available after publication of this work, but is currently excluded for maintaining anonymity of this submission. 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 [4]. The properties of each dataset are described in Table 4.

Twitter. The *PSB 2016 Social Media Shared Task* Twitter dataset (i.e. collected as explained in [37]) is a widely used manually annotated training data for ADR detection. The original dataset contained a total of 2,000 tweet IDs[5]; 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`[6], `DiagnoseMe` [7], or `Bipolar` [8] 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 [21], 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 4.

4.3 Compared Methods

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

- **QuickUMLS [45]:** an approximate dictionary matching algorithm which relies on UMLS concepts, to match the terms in the test sets with UMLS concepts. We used the following setting, mentioned in [45] as having best performance: *Similarity threshold* = 0.9, *Semantic types* = [*SignorSymptom*, *DiseaseorSyndrome*, *Finding*, *NeoplasticProcess*].
- **Cliner [6]:** an open-source natural language processing system for named entity recognition in clinical text. We used the pre-trained model on the patient discharge summaries (i.e.

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

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

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

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

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

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

Table 4: Dataset statistics. LC: labeled training set, UDC: unlabelled set. Number of sentences, words, and unique ADRs.

	LC(Training)			LC(Testing)			UDC	
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

`i2b2 data`). Cliner is able to detect clinical concepts such as *problem* and *treatment* from the clinical text. We only consider the *problem* tag (i.e. which showed better performance compared to considering both *problem* and *treatment* tags).

- **CRF (Baseline)**. The Conditional Random Field Phrase Detection Model [9] trained on the manually annotated training data *LC*. Detailed information about the selected features can be found online [10].
- **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[11]:** 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 combines a forward and a backward RNN and uses pre-trained word embeddings to identify mentions of ADR in Twitter posts. We used the *pre-trained-fixed* setting which treated the word-embedding values as fixed constants (i.e. this is the setting with the highest reported performance). Typically, BLSTM-RNN are expected to perform better than simpler CRF approaches.
- **BLSTM-RNN+VAE (proposed):** For this method, we combined our proposed technique for artificial training data generation 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 expanding the training data for ADR phrase detection, we compare our proposed approach with the following techniques.

- **CRF+SelfTraining [49]:** 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 [27] which is inspired by word2vec [34] to find sentences similar to the labeled content *LC*.

[9] https://github.com/dat/stanford-ner
[10] https://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/ie/NERFeatureFactory.html

4.4 Training

Each dataset is split into training and testing according to Table 4. 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 for both the encoder and the decoder. For labeling the newly generated sentences, we used word embeddings as described in [34]. For Twitter we used pre-trained word embeddings trained on Twitter as described in [14]. 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, learning all word vectors of all terms in the unlabeled corpus.

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 5 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* and *Cliner* are both 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. On the other hand *Cliner* has been trained on a text with different structure than the user-generated content considered in this paper, and its not able to perform well on Twitter and Reddit datasets.

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. As expected, BLSTM-RNN outperforms CRF in Twitter dataset when the training dataset is smaller; note that the model was designed to detect ADRs from the Twitter dataset. To highlight that our method can be combined with any supervised phrase detection technique, we combined our proposed technique with the BLSTM-RNN. 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 [11], the remaining experiments just focus on CRF for training ADR detector.

Table 5: Performance of the different ADR detection techniques on the Twitter and Reddit test sets.

Technique	Precision	Recall	Fscore
QuickUMLS	.47	.34	.39
Cliner	.30	.13	.18
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
Cliner	.14	.33	.20
CRF	**.72**	.47	.57
BLSTM-RNN	.67	.28	.39
CRF+VAE	.69	**.52**	**.60**
BLSTM-RNN+VAE	.63	.29	.40

(b) Reddit

Table 6: 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
Twitter	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)
	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
Reddit	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)
	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

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.

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. 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. $p<0.05$).

When artificially expanding training data, recall is often improved at the cost of precision. This is demonstrated by the performance of CRF+Doc2Vec (Table 6). 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. Overall, with 10 extra sentences generated, our approach is able to achieve average improvements of +.12 and +.08 for Twitter and Reddit dataset respectively.

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[11] achieved by these techniques in Table 6.

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

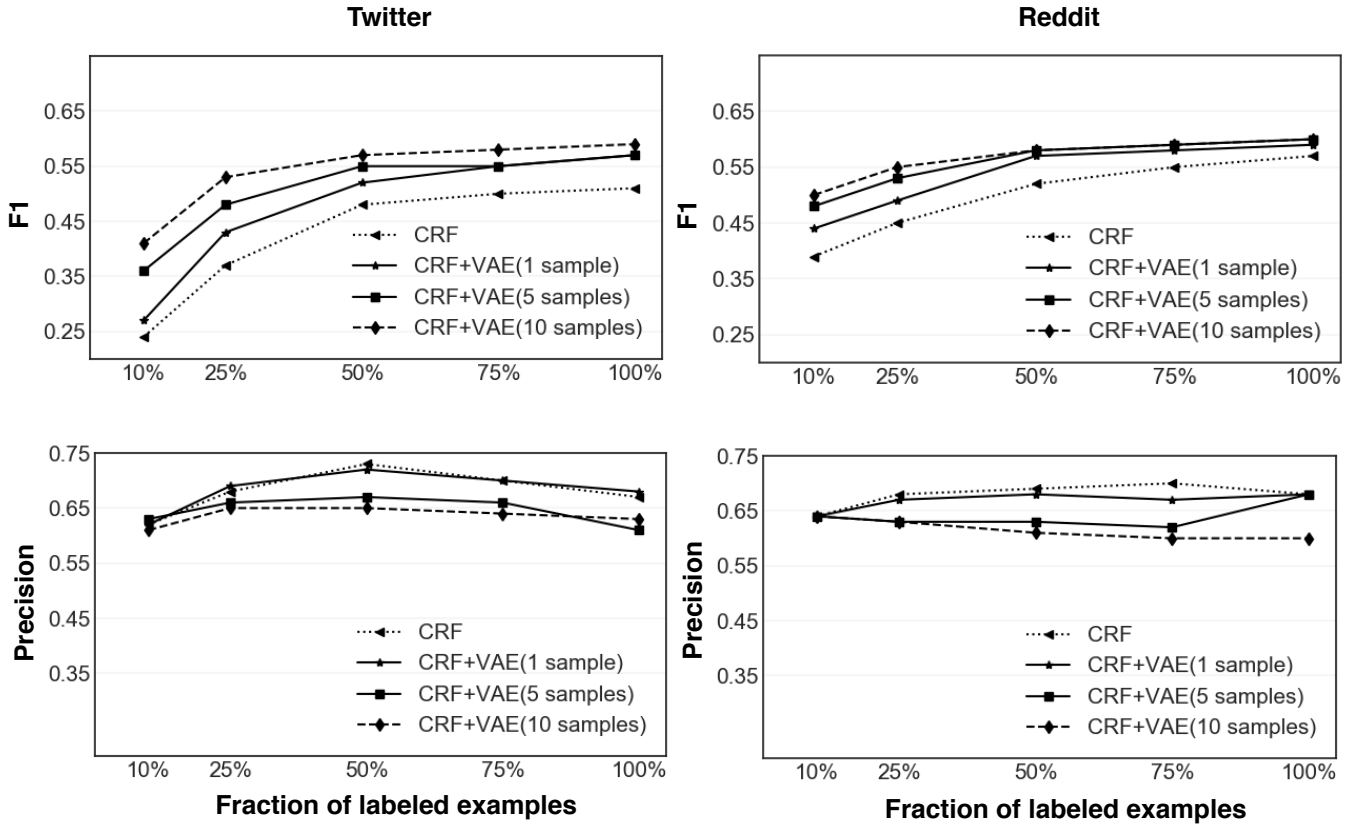

Figure 2: Average *F1* and *Precision* for *CRF* and *CRF+VAE* techniques, trained using different fractions of manually annotated examples and varying number of samples generated using VAE. Tested on the *Twitter* test set (on the left) and on the *Reddit* test set (on the right).

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 [44].

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 false negatives to understand the reasons for these 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 [10]. 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, `depression` commonly occur as ADR as well in other posts, which might be the cause for this error [10]; 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 or BLSTM-RNN as some level of abstraction would be necessary to deal with this.

7 SUMMARY AND CONCLUSIONS

In this paper, we have demonstrated an approach for training sequence taggers for detecting mentions of Adverse Drug Reactions from social media text in a very cost-efficient manner. Detecting ADRs in social data can be used to enrich the overall knowledge on drug effects beyond traditional and expensive sources like clinical trials or extensive consumer and practitioner surveys. However, automatically mining ADR mentions in social media text is hard: the terminology adopted in most social communities makes heavy use of slang or indirect descriptions, is often lacking with respect to grammar and orthography, and in addition is also constantly evolving and differs between communities. This makes the use of established techniques relying on expert-curated dictionaries of consumer health vocabulary or fully-supervised machine learning-based classifiers expensive, and in many cases even prohibits their use. While techniques to lessen the costs of training like distant supervision or bootstrapping can provide some support, their performance has been shown to be limited. To address this issue, we introduced a technique which expands human-labeled training sets with a large number of artificially generated training samples. As our approach is particularity effective on smaller training sets, this can be used to reduce costs of manual annotation for training while still maintaining ADR detection quality. For realizing this goal, we modified Variational Autoencoders in such way that we can generate new realistic artificial training sentences from a given corpus (like Twitter Tweets or Reddit posts) resembling the subset of the corpus for which human annotation are available. Then, we heuristically annotate the new sentences by propagating the labels.

We performed extensive evaluations using an established ADR test and training dataset consisting of Twitter Tweets. Furthermore, in order to investigate a second social community with a vastly different communication and language style we created an own test and training data set from Reddit sub forums, annotating ADRs with the help of professional medical experts. This dataset is publicly available for future research [12]. While our work is evaluated on medical text focusing on ADR, our approach should be effective for any kind of supervised text sequence detection problem (like for example Named Entity Recognition), and exploring it for additional domains is addressed in our future work.

[12]URL removed for anonymity during review process

On both datasets, we could show that our technique for training an ADR detector outperforms other training techniques like self-training, fully supervised training, or embedding-based training set expansion independent of the actual detection algorithm (we evaluated with Conditional Random Fields and BLSTM-RNN-based classifiers). Furthermore, we outperform dictionary-based techniques by a wide margin (F-Score of 0.39 for dictionary-based Cliner, vs. our approach with CRF 0.51, or with BLSTM-RNN 0.77). In particular, we could show that the effect of our training data generation technique is greater the smaller the training data set is. Therefore, the increase in quality we can achieve is used most efficiently to reduce the training costs of ADR detection while maintaining quality (e.g., in our experiments, we can maintain the same quality with an F-Score of 0.53 when using CRF with just 25% of the training data).

However, there is a saturation effect: when sufficient manual training data is available, further artificial data generation has only reduced positive effects. 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. 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 - imitating a typical web user's Tweet behavior seems to be easier for a VAE in comparison. VAEs are known to work more effectively with shorter sentences than with longer ones. In our qualitative evaluation manually inspecting false positives and negatives, we could identify several core problems: sentences using explicit or implicit negations, confusion between drug indications and adverse effects, wrongly annotated test data, and very indirect informal expressions. Nonetheless, our core challenge to reduce training costs for ADR detection is addressed well by our approach.

This work is only one of the initial steps towards automated adverse drug effect analytics on social data. We could show that detecting and extracting ADRs from user generated text containing slang terms can be performed effectively at comparably low costs. However, the next step would be to interpret the semantics of the extracted slang ADRs, and linking them to medical ontologies and taxonomies to allow for further structured analysis.

REFERENCES

[1] 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 (2003), 57–60.

[2] 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*. Association for Computational Linguistics, 1568–1576.

[3] Naomi S Baron. 2010. *Always on: Language in an online and mobile world*. Oxford University Press.

[4] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. 2017. Variational inference: A review for statisticians. *J. Amer. Statist. Assoc.* 112, 518 (2017), 859–877.

[5] 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*. ACM, 92–100.

[6] William Boag, Kevin Wacome, Tristan Naumann, and Anna Rumshisky. 2015. CliNER: A lightweight tool for clinical named entity recognition. *AMIA Joint Summits on Clinical Research Informatics (poster)* (2015).

[7] 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. 10–21.

[8] Suthee Chaidaroon and Yi Fang. 2017. Variational deep semantic hashing for text documents. In *SIGIR*. ACM, 75–84.

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

[10] 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*. International World Wide Web Conferences Steering Committee, 117–126.

[11] 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 (2017), 813–821.

[12] 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*. International World Wide Web Conferences Steering Committee, 1045–1052.

[13] Alexey Dosovitskiy and Thomas Brox. 2016. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*. 658–666.

[14] 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*. 146–153.

[15] Sean Goldberg, Daisy Zhe Wang, and Christan Grant. 2017. A Probabilistically Integrated System for Crowd-Assisted Text Labeling and Extraction. *Journal of Data and Information Quality (JDIQ)* 8, 2 (2017), 10.

[16] 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 (2012), 1010–1021.

[17] Z. Harris. 1954. Distributional Structure. *Word* 10 (1954), 146–162.

[18] 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 (2017), 75–85.

[19] 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*. 877–887.

[20] Ling Jiang and Christopher C Yang. 2015. Expanding consumer health vocabularies by learning consumer health expressions from online health social media. In *International Conference on Social Computing, Behavioral-Cultural Modeling, and Prediction*. Springer, 314–320.

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

[22] 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*. International World Wide Web Conferences Steering Committee, 137–146.

[23] Mayank Kejriwal and Pedro Szekely. 2017. Information Extraction in Illicit Web Domains. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 997–1006.

[24] 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*. ACM, 553–562.

[25] Diederik P Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. *stat* 1050 (2014), 1.

[26] 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*, Vol. 951. 282–289.

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

[28] 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*. Association for Computational Linguistics, 117–125.

[29] Kathy Lee, Sadid A Hasan, Oladimeji Farri, Alok Choudhary, and Ankit Agrawal. 2017. Medical Concept Normalization for Online User-Generated Texts. In *Healthcare Informatics (ICHI), 2017 IEEE International Conference on*. IEEE, 462–469.

[30] 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*. International World Wide Web Conferences Steering Committee, 705–714.

[31] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *WWW*. IW3C2, 689–698.

[32] 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*. Springer, 127–143.

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

[34] 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*. 3111–3119.

[35] 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*. Association for Computational Linguistics, 1003–1011.

[36] Azadeh Nikfarjam and Graciela H Gonzalez. 2011. Pattern mining for extraction of mentions of adverse drug reactions from user comments. In *AMIA Annual Symposium Proceedings*, Vol. 2011. American Medical Informatics Association, 1019.

[37] Azadeh Nikfarjam, Abeed Sarker, Karen OâĂŹconnor, 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 (2015), 671–681.

[38] Michael J Paul and Mark Dredze. 2011. You are what you Tweet: Analyzing Twitter for public health. *Icwsm* 20 (2011), 265–272.

[39] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. 2016. Variational autoencoder for deep learning of images, labels and captions. In *Advances in neural information processing systems*. 2352–2360.

[40] Alexander Ratner, Stephen H Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. *Proceedings of the VLDB Endowment* 11, 3 (2017), 269–282.

[41] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic Backpropagation and Approximate Inference in Deep Generative Models. In *International Conference on Machine Learning*. 1278–1286.

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

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

[44] 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*. 627–637.

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

[46] Akash Srivastava and Charles Sutton. 2017. Autoencoding variational inference for topic models. *In Proceedings of ICLR* (2017).

[47] 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*. ACM, 1733–1738.

[48] 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 (2006), 92.

[49] 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*. Association for Computational Linguistics, 65–73.

[50] Andrew Yates and Nazli Goharian. 2013. ADRTrace: detecting expected and unexpected adverse drug reactions from user reviews on social media sites. In *European Conference on Information Retrieval*. Springer, 816–819.

[51] FM Zanzotto and Marco Pennacchiotti. 2012. Language evolution in social media: a preliminary study. *LINGUISTICA ZERO* (2012).

[52] Biao Zhang, Deyi Xiong, Jinsong Su, Hong Duan, and Min Zhang. 2016. Variational neural machine translation. *CoRR, abs/1502.04623* (2016).

[53] Xiangling Zhang, Yueguo Chen, Jun Chen, Xiaoyong Du, Ke Wang, and Ji-Rong Wen. 2017. Entity Set Expansion via Knowledge Graphs. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1101–1104.