Learning Structured Medical Information from Social Media

Abul Hasan*, Mark Levene, David Weston

Department of Computer Science and Information Systems, Birkbeck, University of London, London WC1E 7HX, United Kingdom.

Abstract

Our goal is to summarise and aggregate information from social media regarding the symptoms of a disease, the drugs used and the treatment effects both positive and negative. To achieve this we first apply a supervised machine learning method to automatically extract medical concepts from natural language text. In an environment such as social media, where new data is continuously streamed, we need a methodology that will allow us to continuously train with the new data. To attain such incremental re-training, a semi-supervised methodology is developed, which is capable of learning new concepts from a small set of labelled data together with the much larger set of unlabelled data. The semi-supervised methodology deploys a conditional random field (CRF) as the base-line training algorithm for extracting medical concepts. The methodology iteratively augments to the training set sentences having high confidence, and adds terms to existing dictionaries to be used as features with the base-line model for further classification. Our empirical results show that the base-line CRF performs strongly across a range of different dictionary and training sizes; when the base-line is built with the full training data the F_1 score reaches the range 84-90%. Moreover, we show that the semi-supervised method produces a mild but significant improvement over the base-line. We also discuss the significance of the potential improvement of the semi-supervised methodology and found that it is significantly more accurate in most cases than the underlying base-line model.

Keywords: social media mining, medical concept extraction, pharmacovigilance, conditional random fields, semi-supervised algorithm.

1. Introduction

Online health forums and social media such as MedHelp [1] and Twitter, often contain experiential information from patients who share symptoms and side-effects of the prescribed treatments. These shared experiences from a group of patients have been proven to be useful for public health monitoring [2, 3]. To further advance such findings, a group of researchers from the University of Pennsylvania has been organising the Social Media Mining for Health Applications (SMM4H) shared task to detect Adverse Drug Reaction (ADR) from tweets [4]. However, as shown in the examples of Figure 1, social media posts contain not just ADR mentions, they can also include other useful information such as a patients' sentiment regarding their medical condition. In this study, we wish to identify not just ADRs but also effects of a drug that may not be the intended therapeutic outcome and indeed might be considered beneficial, hence we use the term side-effect, [5].

In previous work, we built a framework for concept relation extraction using a natural language processing (NLP) methodology by utilising health related concepts augmented with sentiment as expressed in the text [6]. The methodology was rule-based together with lexicon matching. It is possible that terms contained in social media text may not exist in the publicly available dictionaries and ontologies such as Unified Medical Language System (UMLS) [7]. Consequently, recognising concepts from such colloquial text using lexicon matching algorithms often produce poor results [8]. To address this challenge, researchers have applied supervised machine learning methods, which requires manually annotated training data. Recently, an iterative semi-supervised active learning based method was proposed to recognise drugs and their side-effects from twitter data [9] by including human annotators in the training loop to augment representative and diversified labelled data.

Here, we present a semi-supervised methodology based on conditional random fields (CRFs) [10], which classifies tokens in a sentence belonging to one of the categories shown in Table 1. We believe these classes cover the semantics pertaining to the objective of the research. In our previous work, we deployed a lexicon-based classifier with an NLP relationship extraction system [6]. The rules, which were heavily dependent on the lexicon matching, were inferred from the training dataset by manually analysing the text. Some lexicons were publicly available,

*Corresponding author

Email addresses: abulhasan@dcs.bbk.ac.uk (Abul Hasan), mlevene@dcs.bbk.ac.uk (Mark Levene), dweston@dcs.bbk.ac.uk (David Weston)

Examples

T1 I envy you I can take 75[CD] mg[DOSE] of melatonin[D] and never[NG] fall[SYM] asleep[SYM].

T2 I take those mirapex[D] now for Percocet[D] withdrawals[SD].

T3 Roprineral[D] tablets from the doc for restless[SYM] leg[SYM], they are helping[P] me.

M1 Sinemet[D] increases CNS[SYM] dopamine[SYM] which can lead to psychosis[SD].

M2 My sinemet[D] had been worn[N] off[N] for 5[CD] hours[TMCO] I take it every two[CD] to three[CD] hours[TMCO].

M3 My hands[BPOC] feel really[INT] weak[SYM], but I am able[P] to still[INT] function[P].

Figure 1: T1, T2 and T3 are examples from Twitter and M1, M2 and M3 are those from the MedHelp dataset. The class label of a token is given inside a square bracket. The description of labels is listed in Table 1.

and others were curated manually from the training set. Whereas in this work, the aim is to automate, with minimal supervision, the dependency on the labelled data and the manually created lexicon.

First, a small number of posts and tweets are sampled and annotated. The CRF model was trained on a proportion of the sample, and then this model was applied to the unlabelled data iteratively in order to tag and collect highly confident labelled sentences, symptom and side-effect terms. In an online setting, where data becomes available continuously, as the language changes, the semi-supervised methodology would allow us to automate the incorporation of new terms into dictionaries and be able to adapt to the domain changes with minimal human effort. Here we show that within a single disease category, i.e. Parkinson's, such a continuous training process will either improve or maintain the F_1 score. We thus believe that our method has the additional potential to be used across disease categories with minimal effort, and can be scaled to the practical use needed in medical applications.

In contrast to studies in [8, 11], which focused on medical social media, we deal with more classes and extend the self-training technique [12, 13] to enlarge the training dataset within a semi-supervised framework. Moreover, our methodology involves in minimal human supervision as opposed to the study in [9].

Our system architecture is relatively simple, each class label is coupled with a dictionary feature, and in addition MetaMap [7] is used to determine a small number of useful UMLS semantic types from the text.

Accumulating structured information in the form of a dictionary, which is another point of difference with previous research in this area, has direct impact on the prediction of concepts in a supervised classification task. For example, in the recent SMM4H shared task, the KFU-NLP team, combined contextual word embeddings and Bidirectional Encoder Representations from Transformers (BERT) [14] with dictionary features, and achieved the top result in the identifying ADR span task [4]. Other supervised methodologies such as [8] also rely heavily on lexicons for the improvement in the classification task. Thus, we believe that automatic expansion of dictionaries will allow us to perform incremental learning which is a different task from ontology population [15] and expansion of consumer health vocabulary [16].

We make several contributions, as follows:

1. We show that with a small amount of manually labelled training data we obtain very good performance, and this can be achieved using a semi-supervised methodology which add labelled sentences to the training data in an iterative fashion.

2. Our methodology incrementally augments symptom and side-effect dictionaries by collecting the most confident terms classified by the model. To the best of our knowledge no other previous work attempted to collect learnt health related concepts from the unlabelled data and reuse them in the dictionaries.

3. We combine the above contributions to extend the traditional self-training method [12], by sharing the knowledge in the training data and dictionaries so that sentences, which were rejected at an earlier iteration can still be added when terms are correctly classified at a later iteration.

We tested our methodology on two datasets: the first with posts on Parkinson's from MedHelp, and the second with tweets from Twitter. We then evaluated the performance of the semi-supervised methodology on both data sources by using 100 runs with repeated cross validation; see Section 4.1. To compare the models, we have devised a methodology that can detect potential improvement of the semi-supervised algorithm over the base-line.

2. Related work

There is a growing literature relevant to social media mining for health related information [3, 17, 18, 19]. Most previous research focused on the extraction of ADR concepts from user posts and tweets. ADRMine [8] achieved state-of-the-art results in ADR extraction from a publicly available Twitter corpus [1] by the application of CRFs.

[1] http://diego.asu.edu/downloads/publications/ADRMine/download_tweets

Table 1: Class label, description of the class, and the number of words in the class with the percentage inside a bracket are shown separately for MedHelp and Twitter dataset.

Class	Description	Medhelp	Twitter
D	Drug/Treatment	1233(0.93%)	1822(5.59%)
P	Positive polarity	3453(2.6%)	989(3.04%)
N	Negative polarity	4101(3.0%)	1088(3.34%)
NG	Negation	2265(1.7%)	851(2.61%)
PRE	Pre-suppositionals	325(0.24%)	82(0.25%)
INT	Intensifiers	2514(1.89%)	620(1.9%)
SYM	Symptom	5505(4.14%)	1105(3.39%)
SD	Side-effect	756(0.57%)	580(1.78%)
BPOC	Body parts	2475(1.86%)	233(0.72%)
TMCO	Temporal functions	3572(2.69%)	871(2.67%)
CD	Numbers	3347(2.52%)	564(1.73%)
DOSE	Dosage information	206(0.15%)	211(0.65%)
O	Other	103194(77.62%)	23549(72.31%)
Total		132946	32565

The corpus was annotated with the adverse reaction of a drug, the beneficial effects of the drug, and the health condition/symptom experienced by the patient. Around the same time, the CSIRO Adverse Drug Event Corpus (CADEC) [20] was published with annotated data from the Ask a Patient website [21]. Miftahutdinov et al. [11] also applied CRFs to the CADEC corpus to extract drugs, ADRs, symptoms and clinical findings which could be any medical concept that the annotators are unsure of its category [20]. To boost the performance of the CRFs, both systems created word embeddings from unlabelled data by making use of the Word2vec [22] algorithm, and the resultant word vectors were grouped in predefined clusters that are utilised as features. Apart from the health related social media, a CRF was also applied in an incremental active learning framework for the research in [23] to extract medical concepts from clinical text. Extracting information from electronic health records (EHR) has many challenges similar to those of extracting medical information from social media. A detailed description of methodologies used for information extraction from EHR can be found in [24]. Notably, Chen et al. [25] applied active learning to select training samples from EHR using a small set of labelled data. A bidirectional long short-term memory (LSTM) in conjunction with CRF, was proposed to extract clinical concepts from Chinese EHR [26]. The method extracted diagnoses, tests, body parts, symptoms, and treatments.

Recently, Edo-Osagie et al. [27] used self-training and co-training semi-supervised methods to train different binary classifiers for recognising tweets related to asthma. Lee et al. [28] used semi-supervised Convolutional Neural Network to identify Adverse Drug Events from a publicly available Twitter corpus. They gathered unlabelled corpora from various biomedical sources to learn phrase embeddings using dictionaries. They also expanded a health condition dictionary by selecting similar word vectors from an unlabelled corpus. However, this dictionary expansion did not consider an incremental retraining framework for updating the dictionary at each iteration of the semi-supervised methodology. A semi-supervised methodology was also applied by combining Word2vec with Brown clustering [29] features extracted from the unlabelled Spanish and Swedish electronic health records. These features boosted the performances of different base-line models, including a CRF [30]. More recently, a Chinese drug event report corpus was utilised to compare the effectiveness between the CRF and the Bi-LSTM-CRF model in recognising ADR concepts when deployed within a co-training style tri-training [31] methodology [32]. Both CRF and Bi-LSTM-CRF reported achieving comparable performances by leveraging the unlabelled dataset.

Including sentiment related features also improves the performance of CRFs, as shown by Korkontzelos et al. [33]. Moreover, Alhuzali et al. [34] utilised word representations, transferred from a sentiment detection task, in order to classify tweets mentioning ADRs in a Deep Neural Network (DNN) setting. The application of sentiment and lexicon membership features are also widespread in high performing DNN-based systems submitted to various SMM4H shared tasks. For example, Wu et al. [35] utilised these features with character and word embeddings from a combination of various neural network setting to recognise tweets mentioning ADRs; this was the best performing system in the 2018 SMM4H shared task. Although in the most recent SMM4H in 2019, the binary classification task of identifying tweets containing ADRs has been improved significantly by concatenating contextual embeddings using BERT with various lexical and syntactical features, the extraction of ADRs remains a challenging task [4].

Following the success of contextual word embeddings

in different NLP tasks, BioBERT [36] was fine-tuned on biomedical literature using the pre-trained BERT model. This model significantly outperformed state-of-the-art models on various biomedical information extraction tasks [36]. The BERT model was also pre-trained with clinical notes from hospital admissions, named as ClinicalBERT [37]. This model outperformed both BERT and BioBERT on the hospital readmission prediction task. In addition, BioBERT has shown promising performance in normalising ADR concepts from Twitter to a formal medical language [38].

Our research is motivated differently from the tasks described above, since our intention is to summarise health related posts by extracting more concepts than in the above tasks, and to discover the relation among these with the help of sentiment expressed in the text.

3. Materials and Methods

Our overall methodology is shown schematically in Figure 2 and the corresponding pseudo-code of the semi-supervised algorithm is presented in Algorithm 1. In the following subsections, we first describe the data collection and annotation procedures of the text and then describe the methodology in more detail.

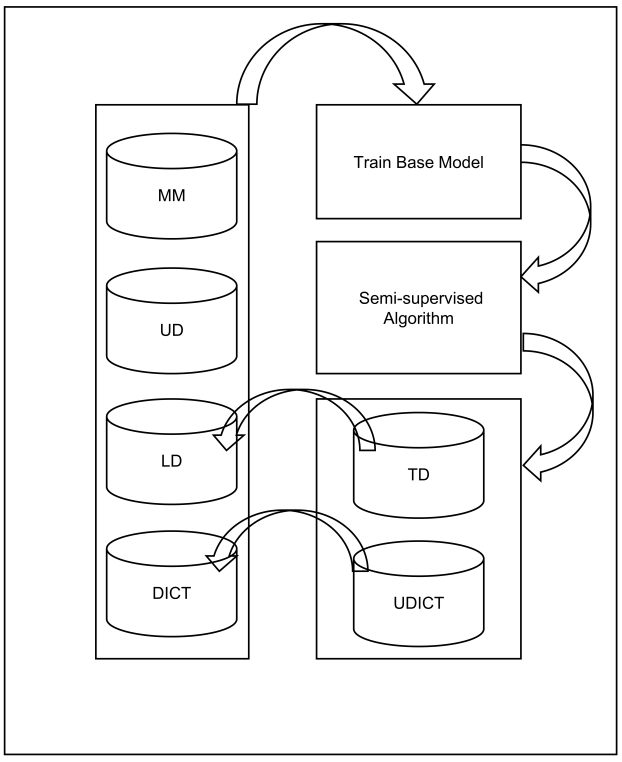


Figure 2: The semi-supervised text processing framework. MM denotes the MetaMap plug-in for UMLS, LD denotes the labelled dataset, UD denotes the unlabelled dataset, $DICT$ denotes the different publicly available dictionaries used, TD denotes the tagged data using the base-line model trained on UD, and $UDICT$ denotes the dictionaries learnt with the semi-supervised algorithm from the unlabelled dataset.

3.1. Data collection and annotation

We collected 1000 user posts discussing Parkinson's disease from the MedHelp forum[1] and the same number of tweets from Twitter. MedHelp posts are collected by using the search system provided by the website [2]. To collect tweets, we used the Twitter search API providing it with a list of known Parkinson's drugs; tweets containing links and photos were excluded from the dataset. The posts and tweets were then anonymised and annotated using the labels shown in Table 1. For the semi-supervised learning algorithm, we collected an additional 4,000 tweets and 15,000 MedHelp posts. The sample size for labelled data was calculated at a 95% confidence interval with 4%-6% error margin, which gave us the size of 600-400 posts/tweets, respectively. Here, the labelled and unlabelled datasets are denoted as LD and UD, respectively; see Figure 2.

3.1.1. Annotation Validation

In order to verify the fidelity of the annotations carried out by the first author, an experiment was conducted using a small subset of the data, where the level of agreement between the annotator and other annotators was measured. Eight annotators were trained by showing them annotated posts explaining drug, symptom and side-effect concepts. Each annotator received an average of 18 posts and tweets from a total of 150. The agreement between the first author and the annotators was calculated using Cohen's κ statistic [39]. The overall agreement reached was 75% after discounting an outlier. Though we achieved a very high level of agreement (81%) for the drug concept, the agreement for the symptom and side-effect concepts were lower at 69% and 74% respectively. However, when we combined the symptom and side-effect into a single class, Cohen's κ reached 75%.

3.2. Training the base-line model

We pre-processed the labelled and unlabelled data, LD and UD respectively, through a built-in feature extraction program in GATE [40] using an NLP pipeline. The NLP pipeline splits the text into sentences and tokens, performs parts-of-speech (POS) tagging, applies lexicons and gazetteers (denoted as $DICT$ in Figure 2) to find the membership of a token and integrate it with MetaMap (denoted as MM in Figure 2) to infer the UMLS semantic class. The labelled dataset, LD, is divided into training, test and validation sets denoted *train*, *test*, and *valid* respectively. We built a model denoted as the *base-line model* by applying a linear-chain CRF [10].

CRFs are a family of undirected graphical model representing conditional distribution and can be applied to calculate the conditional probability of a label sequence given a sentence represented as a sequence of tokens.

Let $X = x_0, \ldots, x_t, \ldots, x_T$ be a sequence of tokens and $Y = y_0, \ldots, y_t, \ldots, y_T$ be their corresponding labels.

[2] https://www.medhelp.org/search?&query=parkinsons

Algorithm 1 Semi-supervised training assuming separate symptom and side-effect classes

INPUT:
LD: Labelled data divided in $train_0$, $test$, and $valid$ sets
$DICT$: Existing dictionaries
UD: Unlabelled data
α: Confidence interval threshold
n: Number of sentences
i_{max}: Maximum number of iterations

$UDICT_0 \leftarrow$ Empty dictionary to store symptom and side-effect predicted from UD
f_0: *base-line model* trained on $train_0$ and $DICT$
$i \leftarrow 0$
repeat
  $TD \leftarrow$ Tag UD by f_i
  $viterbi_i \leftarrow$ The set of n highest viterbi sentences from TD
  $train_i \leftarrow train_{i-1} \cup viterbi_i$
  $UD_i \leftarrow UD_{i-1} - viterbi_i$
  $UDICT_{i+1} \leftarrow UDICT_i \cup$ Symptom and side-effect terms predicted from TD by f_i according to α
  $TD_{mark} \leftarrow$ Mark sentences from TD using $UDICT_i$ % A marked sentence cannot be a most confident sentence
  if $i > 0$ **then**
    $mark_i \leftarrow$ The set of n highest Viterbi sentences from corrected TD_{mark}
    $train_i \leftarrow train_i \cup mark_i$
    $UD_i \leftarrow UD_i - mark_i$
  end if
  $f_{i+1} \leftarrow$ Re train base-line model using $train_i$
  Extract features from UD using $UDICT_i$
  Test f_{i+1} on $valid$ and store F_1 score
until $i < i_{max}$

OUTPUT:
$Semi \leftarrow$ Choose f_i by selecting the maximum F_1

Let $g_k(y_t, y_{t-1}, \mathbf{x}_t)$ be $k = 1, \cdots, K$ feature functions at position t, and $\mathbf{x}_t$ be a vector of extracted features for the token at location t. The conditional probability of a label sequence Y is calculated as follows [10]:

$$P(Y|X) = \frac{1}{Z(X)} \prod_{t=1}^{T} \exp \sum_{k=1}^{K} \lambda_k g_k(y_t, y_{t-1}, \mathbf{x}_t). \quad (1)$$

Here, $Z(X)$ is a normalisation factor, and λ_k is the weight of the kth feature function. The goal of training is to estimate the weights of feature functions from the labelled instances. Our linear chain CRF relies on the following boolean features for the token at position t:

1. Word based features: The token, t, its surrounding context, which includes the previous and next token at $t-1$ and $t+1$ respectively, are mapped to the features similar to those found in [8, 11].
2. Lexicon features: These features represents whether t is a member of one of the following publicly available lexicons. A token can be a member of multiple dictionaries. We make use of the following lexicons:
   (a) The MPQA Subjectivity Lexicon [41] for polarity detection.
   (b) The RXNORM [42] drug lexicon.
   (c) Prepositional, negation and intensifier lexicons built from our previous work [6]. These dictionaries were built using common language usage. Prepositionals flip the polarity of a symptom (e.g., *hardly, barely*), intensifiers are used to intensify the polarity of an expression (e.g., *more*), and negations change the positive polarity to negative and vice versa. A token is matched with all these dictionaries to set these features on/off.
   (d) Symptom dictionary with 180 terms commonly used with Parkinson's disease.
   (e) The SIDER [43] dictionary extended with the terms that occurred frequently in the training sequences.
3. MetaMap mapping: The feature extraction program integrates MetaMap to map tokens to their corresponding semantic classes. We set three features depending on the semantic class the token is mapped to:
   (a) Organic Chemical, *ORCH* and Pharmacologic Substance, *PHSU*,
   (b) Sign or Symptom, *SOSY*, and Disease or Syndrome, *DSYN*, and

(c) Body Part, Organ, or Organ Component, *BPOC*.

4. Rule-based: Our feature extraction program identifies whether:

   (a) The token, t is a member of the built-in temporal gazetteer in GATE, and

   (b) The POS tag of t is *CD* type.

Once features are extracted from the text, the base-line model is trained and tested using the *train* and *test* datasets, respectively by using a Python wrapper [44] for CRFsuite, see [45]. For training , we used the limited-memory BFGS [46] gradient descent technique, which is in-built. The training procedure is set with the default regularisation parameters and a maximum of 100 iterations. See Section 4.1 for a discussion on the distribution of the training and test datasets and procedure for cross-validation followed in this study.
The pre-trained base-line model is applied to the unlabelled data, UD, to obtain the tagged sentences, TD, and new symptom and side-effect terms in $UDICT$, as shown in Figure 2. TD and $UDICT$ are then selected by the semi-supervised algorithm to augment the original training data and the existing dictionaries, *train* and $DICT$, respectively.

3.3. The semi-supervised algorithm

Semi-supervised learning involves using both labelled and unlabelled data to train a model [12]. In our approach we first build a CRF based purely on the labelled training data (the base-line model). This model is then used to predict the labels for the unlabelled training data. We then analyse these predicted labels to identify new words to be included in the dictionaries, which are described in the previous section. We also identify sentences to be included in the labelled training data and removed from the unlabelled set. The CRF is then rebuilt using these updates and the process is repeated until a stopping criterion has been met. We first summarise our semi-supervised method as follows:

1. Train the base-line model using the labelled data.

2. Repeat the following steps until the stopping criteria is satisfied:

   (a) Tag the unlabelled data using the base-line model.

   (b) Identify most confident sentences from the tagged data, add them to the labelled set.

   (c) Identify new symptom and side-effect terms, add them to their relevant dictionary.

   (d) Flag sentences where newly identified symptom and side-effect terms are misclassified.

      - Identify any flagged sentences that subsequently have had their misclassified terms correctly classified. Add these sentences to the labelled set.

   (e) Rebuild the base-line CRF model with the above updates and record the performance on the validation set.

The method is shown in Algorithm 1 and is described in more detail as follows.

Identifying new dictionary terms

We are interested in identifying new symptoms and side-effects, consequently we restrict our search for new terms to these two labels. For each unique word, that does not already exist in a dictionary and is not a known stop word, we collate all the predicted labels. A word will be added to the dictionary corresponding to its most frequent label provided we are confident of that predicted label. Our confidence is measured by estimating the standardised *Wald confidence interval*, CI, [47] at at the 95% level i.e.,

$$CI = \hat{p} - 1.96\sqrt{\hat{p}(1-\hat{p})/n}, \tag{2}$$

where $\hat{p}$ is estimated probability that the word is assigned its most frequent label and n is the number of instances of the word. If the lower bound of the confidence interval, $\hat{p} - CI$, is greater than a threshold (set to 0.5), denoted by α in the Algorithm 1, we proceed to augment the dictionary with this word.

Identifying sentences

The linear-chain CRF produces the best tagged sequence for a sentence with a score similar to the inference probability produced by a Hidden Markov Model known as the *Viterbi* probability [48, 10]. We use this probability to rank sentences and select at most the top five that have a probability above a threshold of 0.9. Ideally we wish to include only one sentence per iteration, however due to computational constraints we increase this figure to five.

There are notable shortcomings of using the Viterbi probability for ranking sentences. First sentences with short length will tend to have a high Viterbi probability and second class label imbalance (the *Other* label dominates) in our data generally results in higher probabilities for sequences labelled with *Other*. To mitigate this bias, we only consider sentences of length greater than 3 that also contain at least one of the drug, symptom or side-effect labels.

The set of highest Viterbi sentences are likely to be similar to sequences that are in the labelled training set [49]. As a result, the augmented training data may lack in diversity. To overcome this, we introduce an additional approach for identifying sentences to include in the labelled training set. As described above, a word will be added to a dictionary provided there is sufficient consistency in the predicted labels. The sentences that contain the word but which have been mislabelled (i.e. not predicted the most frequent label) are flagged. At any subsequent iteration, all flagged sentences are checked to see if any new dictionary word has now been relabelled correctly. These corrected sentences are ranked by their Viterbi probability and the top five are transferred to the labelled training set.

Stopping Criterion and Model Selection

The model parameters of the CRF are recorded after each iteration and the model that provides the highest F_1 score on the validation set is selected as the final model. The maximum number of iterations is fixed at 100. In our experiment, we found that typically the model converges in 30 iterations for the MedHelp data and 15 iterations for the Twitter data. We believe this difference is due to the difference in size of these datasets.

4. Results

We evaluate the performance of the basline and semi-supervised algorithms using precision (P), recall (R) and F-score (F_1), defined as:

$$P = \frac{TP}{TP+FP} \qquad R = \frac{TP}{TP+FN} \qquad F_1 = \frac{2PR}{P+R}$$

True positives (TP), false positives (FP) and false negatives (FN) are calculated by comparing the model's extracted concept with the actual annotation via exact matching at the individual token level. Here we report both macro and micro averaged F_1 scores. Macro scores are computed by considering the score independently for each class and then taking the average, while micro scores are computed by considering all the classes together. The F_1 scores are calculated by averaging over 100 runs of repeated cross validation [50] described next.

4.1. Repeated cross-validation

We permuted our dataset to produce 100 different runs to estimate the average F_1 score, see Table 2. The F_1 score is calculated for each run using a 5-fold cross validation strategy of 20% training and 80% test. This is different from the traditional 80%-20% train-test split because we wish to emulate the situation where there is minimal labelled data. From the test set, which is 80% of the total dataset, one-third is reserved as a validation set to be used with the semi-supervised model.

We make use of fractional training sets which are a subset of the full 20% training set. These smaller subsets have sizes 10%, 25%, 50%, and 75% are used as separate training sets and the F1 score is averaged over them. For example, 10% of the training data yielded 10 disjoint sets from the full training set, which were ran independently to get the F_1 score and then averaged. This evaluation procedure is repeated starting with empty symptom and side-effect dictionaries, and then incrementing the size of the dictionaries by 25%, 50%, 75% and 100%.

Repeated cross-validation is a time and space constrained procedure, which required runs over a network of multiple machines using the Condor [51] distributed batch computing system. More specifically, we employed a network of 135 machines simultaneously, where each machine had 8 to 12 CPU cores, and the algorithm ran over several days.

The base-line CRF produces high macro F_1 scores of 88.90% and 84.3% for MedHelp and Twitter dataset respectively at larger(>=50%) training and dictionary sizes. The score is further improved to 90.90% and 87.2% for MedHelp and Twitter respectively when we combine symptom and side-effect classes to one single class; see the bottom part of Table 2. It is also evident that the improvement of the macro and micro F_1 score by the semi-supervised model is about 1% when we do not use symptom and side-effect dictionaries and the training size is less than 50%. Although, this improvement is not significant at larger dictionary and training sizes, it shows that the performance of the semi-supervised model dominates that of the base-line model. Next, by running an accuracy test on both models, we quantify more precisely how much more accurate is the semi-supervised model in comparison to the base-line model, and whether the difference is significant. We discuss this comparison in Section 4.3. In the next section, we compare our results with some of the previous studies.

4.2. Comparison with related work

The proposed semi-supervised model builds on our previous work [6]. Our method is not directly comparable with other methods in the literature due to it having different objectives (see Section 1). The ADRmine system [8] achieved an F_1 score of 82.1% and 72.1%, on DailyStrength [3] and on Twitter, respectively, for an ADR detection task. Miftahutdinov et al. [11] attained 79.9% for a multi-label classification task using the CADEC corpus. For a discussion, related to the objectives and the methodologies utilised by these two related works, see Section 2. Recently, in the 2019 SMM4H shared task [4] for ADR detection from Twitter, the KFU NLP team [38] reached the best F_1 score of 65.8% in the competition. The team reported to have used the readily available BioBERT-CRF implementation from [36] with standard parameters deployed for BERT-based models. The results from these studies suggest that our proposed semi-supervised model's performance is competitive, see Table 2.

4.3. Comparing the base-line and semi-supervised models

The difference in performance between the base-line and the semi-supervised models is small, to investigate this difference further we constructed a 2×2 contingency table, as shown in Table 4. Here, X_{11} denotes the total count when base-line and semi-supervised models both predicted a concept correctly, whereas, X_{12} represents the number of times the base-line model predicted a concept correctly but the semi-supervised did not. On the other hand, X_{21} is the total count of correct prediction by the semi-supervised model when the base-line model was incorrect. Finally,

[3] http://www.dailystrength.org/

Table 2: Average F_1 scores are calculated omitting the *Other* class. Averages of 100 runs of repeated cross-validation across different dictionary sizes are shown. Here, *Base* and *Semi* denote the results from the base-line and semi-supervised models, respectively.

All classes													
dataset	Dictionary Size	Macro Averages						Micro Averages					
		Training Size						Training Size					
		10%		25%		>=50%		10%		25%		>=50%	
		Base	Semi	Base	Semi	Base	Semi	Base	Semi	Base	Semi	Base	Semi
MedHelp	0%	0.773	0.786	0.839	0.844	0.873	0.875	0.815	0.823	0.846	0.852	0.875	0.878
	25%	0.797	0.807	0.859	0.863	0.887	0.888	0.831	0.836	0.86	0.863	0.882	0.883
	>=50%	0.800	0.811	0.863	0.867	0.889	0.891	0.834	0.838	0.863	0.866	0.884	0.886
Twitter	0%	0.597	0.606	0.732	0.738	0.809	0.813	0.723	0.728	0.790	0.793	0.838	0.840
	25%	0.629	0.640	0.773	0.780	0.841	0.844	0.742	0.749	0.813	0.817	0.856	0.858
	>=50%	0.631	0.643	0.775	0.782	0.843	0.846	0.745	0.751	0.815	0.819	0.858	0.860
All classes after combining symptom and side-effect to one single class													
MedHelp	0%	0.822	0.833	0.880	0.885	0.906	0.908	0.827	0.833	0.858	0.863	0.886	0.888
	25%	0.832	0.841	0.888	0.891	0.909	0.910	0.845	0.848	0.871	0.872	0.890	0.892
	>=50%	0.833	0.842	0.889	0.892	0.909	0.911	0.848	0.850	0.873	0.875	0.892	0.894
Twitter	0%	0.645	0.656	0.786	0.794	0.859	0.862	0.734	0.74	0.802	0.807	0.854	0.856
	25%	0.671	0.682	0.807	0.814	0.871	0.874	0.773	0.779	0.834	0.837	0.873	0.875
	>=50%	0.673	0.684	0.808	0.815	0.872	0.875	0.777	0.783	0.837	0.840	0.876	0.877

Table 3: Accuracy Test: Scores are calculated omitting the *Other* class. Averages of 100 runs of repeated cross-validation across different dictionary sizes are shown. Here, *Base* and *Semi* denote the result from the base-line and semi-supervised models, respectively.

All classes													
dataset	Dictionary Size	Macro Averages						Micro Averages					
		Training Size						Training Size					
		10%		25%		>=50%		10%		25%		>=50%	
		Base	Semi	Base	Semi	Base	Semi	Base	Semi	Base	Semi	Base	Semi
MedHelp	0%	0.968	3.457	0.818	2.031	0.661	1.317	0.078	0.287	0.067	0.169	0.054	0.110
	25%	0.815	2.337	0.697	1.503	0.658	1.151	0.066	0.190	0.057	0.124	0.053	0.096
	>=50%	0.799	2.304	0.711	1.534	0.666	1.177	0.065	0.187	0.058	0.126	0.054	0.099
Twitter	0%	1.164	2.362	1.242	2.237	0.955	1.557	0.348	0.688	0.375	0.675	0.29	0.463
	25%	1.360	3.003	1.160	2.519	0.872	1.515	0.408	0.887	0.347	0.743	0.263	0.453
	>=50%	1.260	2.885	1.455	2.635	0.937	1.442	0.376	0.846	0.444	0.791	0.286	0.430
All classes after combining symptom and side-effect to one single class													
MedHelp	0%	0.806	2.665	0.564	1.582	0.429	0.942	0.060	0.202	0.042	0.12	0.032	0.072
	25%	0.621	1.915	0.440	1.162	0.417	0.817	0.046	0.142	0.033	0.088	0.031	0.062
	>=50%	0.614	1.910	0.426	1.162	0.414	0.811	0.046	0.141	0.032	0.087	0.031	0.062
Twitter	0%	1.365	3.151	1.336	2.681	0.881	1.457	0.289	0.754	0.247	0.552	0.174	0.306
	25%	1.127	2.995	1.091	1.952	0.659	1.156	0.385	0.88	0.384	0.783	0.25	0.419
	>=50%	1.031	2.781	0.939	2.018	0.616	1.091	0.299	0.811	0.282	0.557	0.187	0.314

the cell containing X_{22} represents number of times both models' predictions were incorrect. If N is the number of tokens in the test set, then the accuracy percentage for the semi-supervised model over the base-line model is $100 \times X_{21}/N$, and similarly the percentage of accuracy for the base-line over the semi-supervised model is $100 \times X_{12}/N$. To assess the significance of improvement, we computed the χ^2 value for 1 degree of freedom by making use of X_{12} and X_{21}, which is known as McNemar's non-parametric test [52].

We ran the accuracy test along with the repeated cross validation strategy described above, and the calculated average macro and micro percentages are shown in Table 3. In the case of micro average accuracy, we considered all the tokens in the test set by ignoring their class labels. To calculate the macro average accuracy, the score is considered separately for all the classes and then averaged. The result, shown in Table 3, suggests that the semi-supervised model is generally 1-2% more accurate than the base-line model at every division of dictionary and train sizes over 100 runs. This implies that the semi-supervised model always improves the prediction of base-line model. We now discuss the significance of this improvement by the semi-supervised model.

5. Discussion

In order to compute the statistical significance of a possible improvement of the semi-supervised model over the

Table 4: Contingency table template for comparing accuracy between the semi-supervised and the base-line model.

	Semi-supervised model		
base-line model	Correct	Incorrect	Total
Correct	X_{11}	X_{12}	$X_{1,}$
Incorrect	X_{21}	X_{22}	$X_{2,}$
Total	$X_{,1}$	$X_{,2}$	N

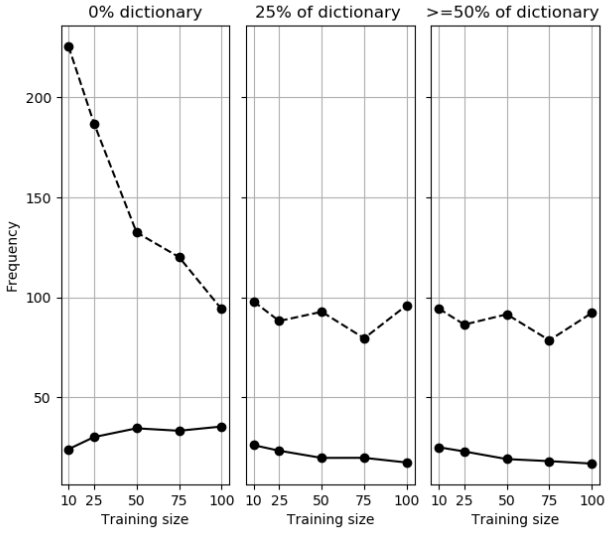
base-line model we made use of the McNemar's test, as described above, with respect to the symptom and side-effect classes and the combined symptom and side-effect class. The results show that the difference is significant for the symptom class for all dictionary and training sizes. Regarding the side-effect class for Twitter, although the semi-supervised model performed better than the base-line model, it is not generally significant, probably due to the imbalance between the side-effect and symptom classes; in particular the side-effect class is much smaller in size than the symptom class, which gives a priori preference to the symptom class over the side-effect during the classification process. Moreover, for MedHelp, the symptom class is also larger than the side-effect class, even greater than in Twitter. In this case it seems that the misclassification of side-effects as symptoms by the semi-supervised model is accentuated further due to this large class imbalance.
In Figures 3, 4 and 6, we have shown the comparison between the models in predicting symptom, side-effect and the combined symptom and side-effect classes at different dictionary sizes. As described earlier, the averages of X_{12} and X_{21} from 100 repeated cross validated runs are plotted on y-axis against different training sizes on x-axis. These averages are used as input for McNemar's test. In addition, we considered the minimum of both X_{12} and X_{21}, which calculates a *conservative* estimate for the said significance test.

5.1. Symptom prediction

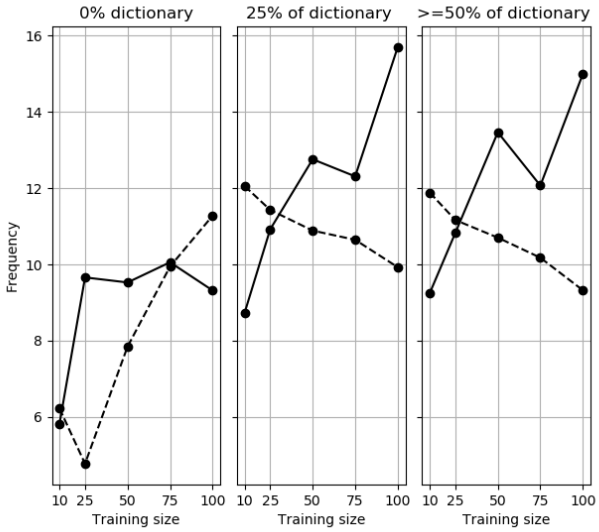
For the MedHelp dataset the semi-supervised model correctly predicted, on average, 100 symptom terms more than the base-line model; see Figure 3a. This difference is significant for the symptom class at the 95% confidence level using Mcnemar's test; the test makes use of the conservative estimate as described above. Although for the symptom class, the margin of difference for Twitter dataset is smaller, as seen in Figure 4a, this difference is also significant. In Figure 5a we see an example of improvement over the base-line model. In this case, the semi-supervised model correctly recognises *shakes* as symptom while the base-line model classifies *shakes* as *Other*.

5.2. Side-effect prediction

In the case of the MedHelp dataset, we found that the accuracy of predicting side-effect degrades slightly. In general many symptom and side-effect terms are common in



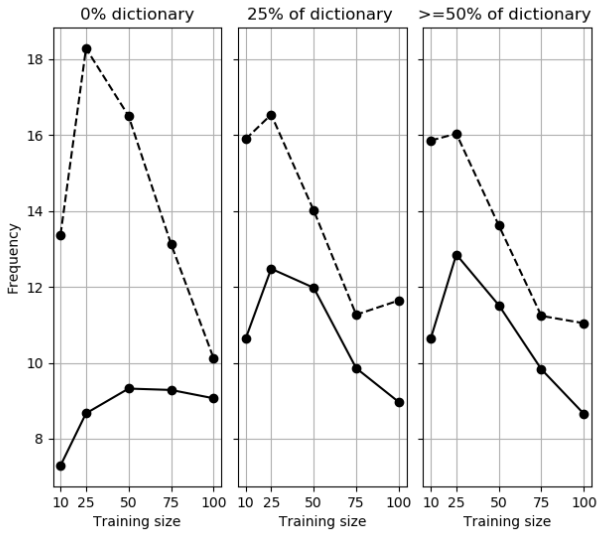
(a) Symptom class



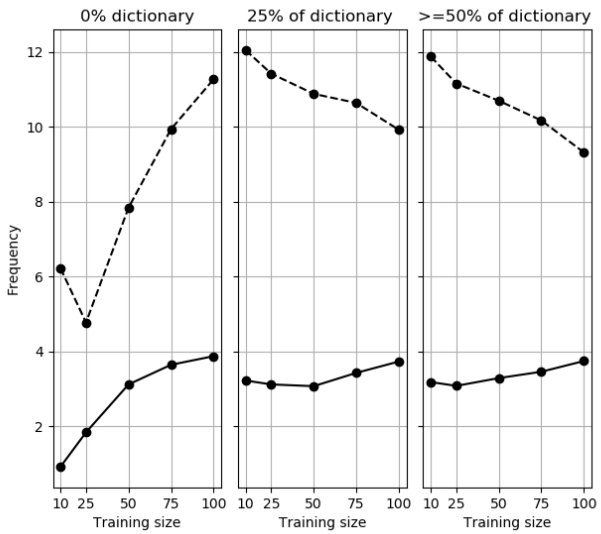
(b) Side-effect class

Figure 3: MedHelp: Comparison of base-line and semi-supervised models in predicting (a) symptom and (b) side-effect classes by using MedHelp dataset. Lines and dots represent base-line and semi-supervised model, respectively.

both respective dictionaries creating ambiguity and possible misclassification. The cause of the ambiguity is most likely due to symptom and side-effect often appearing in common contexts. We found that in such cases, even a human annotator may find it difficult to distinguish between these classes. The large class imbalance for MedHelp, as shown in Table 1, causes the transition probabilities of symptom terms to be higher than those of side-effect terms. Thus during test phase, the semi-supervised model gives priority to symptom over side-effect. As a consequence, the semi-supervised model collects more symptom



(a) Symptom class



(b) Side-effect class

Figure 4: Twitter: Comparison of base-line and semi-supervised models in predicting (a) symptom and (b) side-effect classes using Twitter dataset. Lines and dots represent base-line and semi-supervised model, respectively.

Example of an improvement

1. The worst_{N} part is that it has $\text{affected}_{\text{N}}$ my $\text{left}_{\text{O-BPOC}}$ $\text{hand}_{\text{O-BPOC}}$ more_{INT} and my $\text{pinky}_{\text{O-BPOC}}$ and $\text{ring}_{\text{O-BPOC}}$ $\text{finger}_{\text{O-BPOC}}$ have $\text{really}_{\text{INT}}$ fast_{INT} $\textbf{shakes}_{\textbf{SYM}}$.
2. Base : [‘O’, ‘N’, ‘O’, ‘O’, ‘O’, ‘O’, ‘O’, ‘O’, ‘O’, ‘O-BPOC’, ‘O-BPOC’, ‘INT’, ‘O’, ‘O’, ‘O-BPOC’, ‘O’, ‘O-BPOC’, ‘O-BPOC’, ‘O’, ‘INT’, ‘INT’, **‘O’**]
3. Semi : [‘O’, ‘N’, ‘O’, ‘O’, ‘O’, ‘O’, ‘O’, ‘O’, ‘O’, ‘O-BPOC’, ‘O-BPOC’, ‘INT’, ‘O’, ‘O’, ‘O-BPOC’, ‘O’, ‘O-BPOC’, ‘O-BPOC’, ‘O’, ‘INT’, ‘INT’, **‘SYM’**]

(a) Example of an improvement, where *shakes* was correctly classified by the semi-supervised model as *SYM*.

Example of a misclassification

1. The side_{N} $\text{effects}_{\text{N}}$ were noted to be mild_{P} and included $\text{diarrhea}_{\text{SD}}$ neck_{SD} pain_{SD} and dry_{SD} mouth_{SD} .
2. Base : [‘O’, ‘N’, ‘N’, ‘O’, ‘O’, ‘O’, ‘O’, ‘P’, ‘O’, ‘O’, ‘SD’, ‘SD’,**‘SD’**, ‘O’, ‘SYM’, ‘SYM’, ‘O’]
3. Semi : [‘O’, ‘N’, ‘N’, ‘O’, ‘O’, ‘O’, ‘O’, ‘P’, ‘O’, ‘O’, ‘SD’, ‘SD’, **‘SYM’**, ‘O’, ‘SYM’, ‘SYM’, ‘O’]

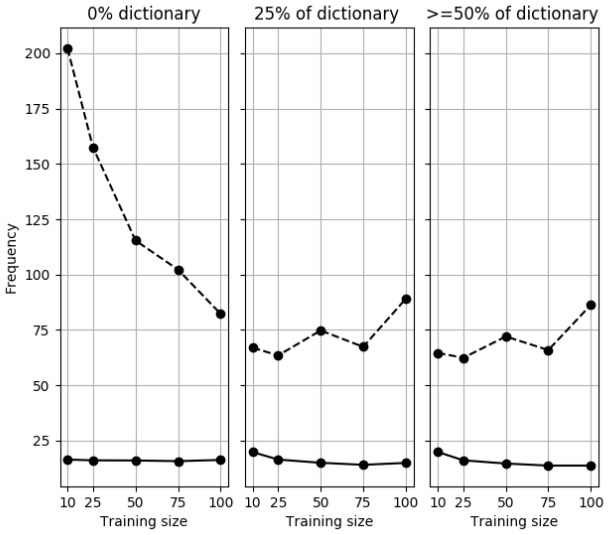
(b) Example of a misclassification by the semi-supervised model, where *pain* was incorrectly classified as *SYM* instead of *SD*.

Figure 5: Examples of (a) an improvement and (b) a misclassification made by the semi-supervised model. Here, at 1, we have a sentence with annotated labels in the subscript, at 2 and 3 the predicted labels by the base-line and semi-supervised models are, respectively, given. The boldface letters signal either an improvement or a misclassification.

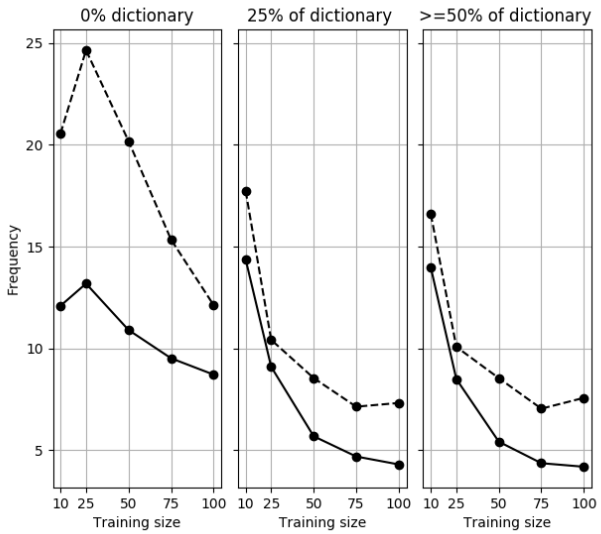
terms than side-effects and the misclassification of side-effect as symptom occurs occasionally. In Figure 5b, we can see this in action; the semi-supervised model misclassified the term *pain* as symptom, denoted by *SYM*, where as the underlying base-line model classified it correctly as side-effect, denoted as *SD*. The term *pain*, exists simultaneously in the symptom and side-effect dictionaries. Moreover, as the transition probability is higher for symptom classes, the model marginally predicts an incorrect label. However, this problem is not present in case of Twitter dataset as the symptom classes are only about twice more in size than side-effect. Though, in the case of Twitter, the improvement over the base-line model is not significant in the conservative estimate; in the average case, it is significant except at larger dictionary and training sizes. Next, we combined the symptom and side-effect terms into a single term and reran the whole procedure again. The result of this process is described next.

5.3. Combining symptom and side-effect

When we combine the symptom and side-effect classes into a single class, the F_1 score for the base-line model improved significantly for both datasets, even more for the semi-supervised model; see the bottom part of Table 2. McNemar's test shows a significant difference between the models, and the semi-supervised model is generally more



(a) MedHelp



(b) Twitter

Figure 6: Comparison of base-line and semi-supervised models in predicting after combining symptom and side-effect classes in (a) MedHelp and (b) Twitter dataset. Lines and dots represent base-line and semi-supervised model, respectively.

accurate than the base-line, see Table 3. For MedHelp the prediction of the combined symptom and side-effect class by the semi-supervised model is significantly better than that of the base-line model; see Figure 6a. Although the experiment with Twitter dataset shows slightly less improvement, it is also significant in most cases; see Figure 6a, except at the 50% of training and dictionary sizes in conservative estimate. In the average case, the semi-supervised improves over the base-line model significantly for all cases; see Figure 6b.

6. Conclusion

We have proposed a semi-supervised algorithm, designed to enhance an underlying pre-trained base-line model, for extracting health related concepts from social media. This algorithm improves on the base-line model when a small amount of labelled data is available, this means that manual annotation can be kept to a minimum. Central to our approach is a procedure for automatically expanding dictionaries of medical concepts, in particular, symptoms and side-effects. These additional words/phrases are also used to identify a diversified set of sentences with which to augment the training data. Although the performance of our method does not drastically improve on that of the base-line model, this process has the potential to be applied in practical usage where the language changes continuously. In such a setting the proposed model will be able to adapt to the changes, as is shown in our experiments. Our future work will involve investigating possible improvement to predictions of diversified training data by utilising word embeddings [22].

References

[1] MedHelp, `https://www.medhelp.org/`, accessed: 2018-06-18.

[2] M. J. Paul, M. Dredze, You are what you tweet: Analyzing twitter for public health, in: Fifth International AAAI Conference on Weblogs and Social Media, July 2011, pp. 265–272.

[3] R. Leaman, L. Wojtulewicz, R. Sullivan, A. Skariah, J. Yang, G. Gonzalez, 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, July 2010, pp. 117–125.

[4] D. Weissenbacher, A. Sarker, A. Magge, A. Daughton, K. O'Connor, M. Paul, G. Gonzalez, Overview of the fourth social media mining for health (smm4h) shared tasks at acl 2019, in: Proceedings of the Fourth Social Media Mining for Health Applications (# SMM4H) Workshop & Shared Task, Aug 2019, pp. 21–30.

[5] I. R. Edwards, J. K. Aronson, Adverse drug reactions: definitions, diagnosis, and management, The lancet 356 (9237) (2000) 1255–1259.

[6] A. Hasan, M. Levene, D. J. Weston, Natural language analysis of online health forums, in: International Symposium on Intelligent Data Analysis, Springer, Oct 2017, pp. 125–137.

[7] O. Bodenreider, The unified medical language system (UMLS): integrating biomedical terminology, Nucleic Acids Res. 32 (1) (Jan 2004) 267–270.

[8] A. Nikfarjam, A. Sarker, K. O'Connor, R. Ginn, G. Gonzalez, Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features, J Am Med Inform Assoc. 22 (3) (Mar 2015) 671–681.

[9] S. Burkhardt, J. Siekiera, J. Glodde, M. A. Andrade-Navarro, S. Kramer, Towards identifying drug side effects from social media using active learning and crowd sourcing, in: Pacific Symposium of Biocomputing (PSB), Vol. 2020, World Scientific, 2020.

[10] C. Sutton, A. McCallum, An introduction to conditional random fields, Found. Trends Mach. Learn. 4 (4) (Aug 2012) 267–373.

[11] Z. Miftahutdinov, E. Tutubalina, A. Tropsha, Identifying disease-related expressions in reviews using conditional random fields, in: Proceedings of International Conference on Computational Linguistics and Intellectual Technologies Dialog, Vol. 1, Jun 2017, pp. 155–166.

[12] X. Zhu, A. B. Goldberg, Introduction to semi-supervised learning, Synthesis lectures on artificial intelligence and machine learning 3 (1) (Jun 2009) 1–130.
[13] J. E. van Engelen, H. H. Hoos, A survey on semi-supervised learning, Machine Learning (2019) 1–68.
[14] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Jun 2019, pp. 4171–4186.
[15] R. Xu, K. Supekar, A. Morgan, A. Das, A. Garber, Unsupervised method for automatic construction of a disease dictionary from a large free text collection, in: AMIA annual symposium proceedings, Vol. 2008, American Medical Informatics Association, 2008, p. 820.
[16] G. Gu, X. Zhang, X. Zhu, Z. Jian, K. Chen, D. Wen, L. Gao, S. Zhang, F. Wang, H. Ma, et al., Development of a consumer health vocabulary by mining health forum texts based on word embedding: Semiautomatic approach, JMIR medical informatics 7 (2) (2019) e12704.
[17] S. Gupta, D. L. MacLean, J. Heer, C. D. Manning, Induced lexico-syntactic patterns improve information extraction from online medical forums, J Am Med Inform Assoc 21 (5) (Jun 2014) 902–909.
[18] H. Sampathkumar, X.-w. Chen, B. Luo, Mining adverse drug reactions from online healthcare forums using hidden markov model, BMC Med Inform Decis Mak 14 (1) (Dec 2014) 91.
[19] T. Huynh, Y. He, A. Willis, S. Rüger, Adverse drug reaction classification with deep neural networks, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Dec 2016, pp. 877–887.
[20] S. Karimi, A. Metke-Jimenez, M. Kemp, C. Wang, Cadec: A corpus of adverse drug event annotations, J Biomed Inform X 55 (Jun 2015) 73–81.
[21] AskAPatient, `https://www.askapatient.com/`, accessed: 2019-07-11.
[22] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in Neural Information Processing Systems, May 2013, pp. 3111–3119.
[23] M. Kholghi, L. Sitbon, G. Zuccon, A. Nguyen, Active learning: a step towards automating medical concept extraction, J Am Med Inform Assoc 23 (2) (Aug 2015) 289–296.
[24] F. Liu, C. Weng, H. Yu, Advancing clinical research through natural language processing on electronic health records: traditional machine learning meets deep learning, in: Clinical Research Informatics, Springer, 2019, pp. 357–378.
[25] Y. Chen, R. J. Carroll, E. R. M. Hinz, A. Shah, A. E. Eyler, J. C. Denny, H. Xu, Applying active learning to high-throughput phenotyping algorithms for electronic health records data, J Am Med Inform Assoc. 20 (e2) (2013) e253–e259.
[26] Y. Zhang, X. Wang, Z. Hou, J. Li, Clinical named entity recognition from chinese electronic health records via machine learning methods, JMIR medical informatics 6 (4) (2018) e50.
[27] O. Edo-Osagie, G. Smith, I. Lake, O. Edeghere, B. De La Iglesia, Twitter mining using semi-supervised classification for relevance filtering in syndromic surveillance, PloS one 14 (7) (2019).
[28] K. Lee, A. Qadir, S. A. Hasan, V. Datla, A. Prakash, J. Liu, O. Farri, Adverse drug event detection in tweets with semi-supervised convolutional neural networks, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 705–714.
[29] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, J. C. Lai, Class-based n-gram models of natural language, Computational linguistics 18 (4) (1992) 467–479.
[30] A. Perez, R. Weegar, A. Casillas, K. Gojenola, M. Oronoz, H. Dalianis, Semi-supervised medical entity recognition: A study on spanish and swedish clinical corpora, Journal of biomedical informatics 71 (2017) 16–30.
[31] Z.-H. Zhou, M. Li, Tri-training: Exploiting unlabeled data using three classifiers, IEEE Transactions on Knowledge & Data Engineering (11) (2005) 1529–1541.
[32] Y. Chen, C. Zhou, T. Li, H. Wu, X. Zhao, K. Ye, J. Liao, Named entity recognition from chinese adverse drug event reports with lexical feature based bilstm-crf and tri-training, Journal of biomedical informatics 96 (2019) 103252.
[33] I. Korkontzelos, A. Nikfarjam, M. Shardlow, A. Sarker, S. Ananiadou, G. H. Gonzalez, Analysis of the effect of sentiment analysis on extracting adverse drug reactions from tweets and forum posts, J Biomed Inform X 62 (Aug 2016) 148–158.
[34] H. Alhuzali, S. Ananiadou, Improving classification of adverse drug reactions through using sentiment analysis and transfer learning, in: Proceedings of the 18th BioNLP Workshop and Shared Task, Aug 2019, pp. 339–347.
[35] C. Wu, F. Wu, J. Liu, S. Wu, Y. Huang, X. Xie, Detecting tweets mentioning drug name and adverse drug reaction with hierarchical tweet representation and multi-head self-attention., in: Proceedings of the 2018 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop and Shared Task, Oct 2018, pp. 34–37.
[36] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, Biobert: a pre-trained biomedical language representation model for biomedical text mining, J Am Med Inform Assoc. 36 (4) (2020) 1234–1240.
[37] E. Alsentzer, J. Murphy, W. Boag, W.-H. Weng, D. Jindi, T. Naumann, M. McDermott, Publicly available clinical bert embeddings, in: Proceedings of the 2nd Clinical Natural Language Processing Workshop, 2019, pp. 72–78.
[38] Z. Miftahutdinov, I. Alimova, E. Tutubalina, Kfu nlp team at smm4h 2019 tasks: Want to extract adverse drugs reactions from tweets? bert to the rescue, in: Proceedings of the Fourth Social Media Mining for Health Applications (# SMM4H) Workshop & Shared Task, 2019, pp. 52–57.
[39] I. H. Witten, E. Frank, M. A. Hall, C. J. Pal, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2016.
[40] H. Cunningham, D. Maynard, K. Bontcheva, Text processing with gate, CA: Gateway Press, 2011.
[41] T. Wilson, J. Wiebe, P. Hoffmann, Recognizing contextual polarity in phrase-level sentiment analysis, in: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), Association for Computational Linguistics, 2005, pp. 347–354.
[42] RXNORM, `https://www.nlm.nih.gov/research/umls/rxnorm/`, accessed: 2019-03-18.
[43] M. Kuhn, I. Letunic, L. J. Jensen, P. Bork, The sider database of drugs and side effects, Nucleic Acids Res 44 (1) (2015) 1075–1079.
[44] python crfsuite, `https://python-crfsuite.readthedocs.io/en/latest/`, accessed: 2018-03-14.
[45] N. Okazaki, Crfsuite: a fast implementation of conditional random fields (crfs) (2007).
URL `http://www.chokkan.org/software/crfsuite/`
[46] J. Nocedal, Updating quasi-newton matrices with limited storage, Math Comput 35 (151) (1980) 773–782.
[47] S. Wallis, Binomial confidence intervals and contingency tests: mathematical fundamentals and the evaluation of alternative methods, J Quant Linguist 20 (3) (Aug 2013) 178–208.
[48] L. R. Rabiner, A tutorial on hidden markov models and selected applications in speech recognition, Proceedings of the IEEE 77 (2) (Feb 1989) 257–286.
[49] K. Clark, M.-T. Luong, C. D. Manning, Q. V. Le, Semi-supervised sequence modeling with cross-view training, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 1918–1925.
[50] G. Vanwinckelen, H. Blockeel, On estimating model accuracy with repeated cross-validation, in: Proceedings of the 21st Belgian-Dutch Conference on Machine Learning, 2012, pp. 39–44.
[51] D. Thain, T. Tannenbaum, M. Livny, Distributed computing in

practice: the condor experience, Concurrency and computation: practice and experience 17 (2-4) (2005) 323–356.
[52] J. D. Gibbons, S. Chakraborti, Nonparametric statistical inference, Berlin Heidelberg: Springer, 2011.