Towards identifying drug side effects from social media using active learning and crowd sourcing

Sophie Burkhardt, Julia Siekiera, Josua Glodde, Miguel A. Andrade-Navarro, and Stefan Kramer
[1]*Department of Computer Science, Johannes Gutenberg University, Mainz, 55128, Germany*
[2]*Department of Biology, Johannes Gutenberg University, Institute of Molecular Biology, Mainz, 55128, Germany*

Motivation: Social media is a largely untapped source of information on side effects of drugs. Twitter in particular is widely used to report on everyday events and personal ailments. However, labeling this noisy data is a difficult problem because labeled training data is sparse and automatic labeling is error-prone. Crowd sourcing can help in such a scenario to obtain more reliable labels, but is expensive in comparison because workers have to be paid. To remedy this, semi-supervised active learning may reduce the number of labeled data needed and focus the manual labeling process on important information.
Results: We extracted data from Twitter using the public API. We subsequently use Amazon Mechanical Turk in combination with a state-of-the-art semi-supervised active learning method to label tweets with their associated drugs and side effects in two stages. Our results show that our method is an effective way of discovering side effects in tweets with an improvement from 53% F-measure to 67% F-measure as compared to a one stage work flow. Additionally, we show the effectiveness of the active learning scheme in reducing the labeling cost in comparison to a non-active baseline. **Contact:** burkhardt@informatik.uni-mainz.de
Availability: Code and data will be published on `https://github.com/kramerlab`.

Keywords: Active learning; Crowd sourcing; Side effects

1. Introduction

Adverse drug reactions (ADRs) are one of the leading causes of death.[1] Frequently, drugs are pulled off the market due to unforeseen side effects. It is therefore important to monitor the effects of drugs even after they have been approved and put on the market. Social media is an important source of information for this task that is difficult to exploit.

Labeling data from social media is difficult because the data is typically noisy due to the frequent use of colloquial language, abbreviations, emoticons, misspellings and grammatical mistakes. Additionally, there is only a small amount of labeled training data publicly available that covers a limited number of drugs and side effects.

Noisy twitter data is difficult to annotate automatically and manual labeling is expensive. Therefore we propose to use semi-supervised active learning in combination with crowd sourcing the annotation process via Amazon Mechanical Turk (MTurk). In active learning (AL), the data that is to be annotated is actively selected. In this way, the classification algorithm is presented with more informative training examples, which helps to speed up training and

save annotation cost. A common way of doing this is to select the training examples that the classifier is most uncertain about. However, it is also important to consider the representativeness of selected examples so as to not query too many outliers. Semi-supervised learning takes advantage of the unlabeled data to learn the distribution of the data, which may improve classification.

MTurk is a service provided by Amazon which helps to crowd source simple repetitive tasks, which nevertheless need to be completed by a human, to workers all over the world. E.g., one task could be to label one tweet with a side effect and for each tweet, a worker is paid a certain, small amount of money. The combination of MTurk and active learning was first proposed by Laws *et al.*,[2] who applied this method on named entity recognition and sentiment classification using a simple pool-based uncertainty active learning scheme. Up to now, we are not aware of any work that uses MTurk and active learning for the identification of side effects on social media.

A relatively small number of tweets was annotated for the TwiMed[3] dataset, where annotators are trained professionals. On the other hand, an automatic labeling approach was taken by Eshleman and Singh,[4] where the graph topology of a bipartite graph is explored. In this approach, the side effects and drugs are nodes in a graph and a connection between a drug and a side effect signifies that the drug has the specific side effect. The labeling in this work is done automatically using a word matching tool called MetaMap.[5] While the first approach is slow and requires expensive experts, the second approach is error-prone and noisy. We aim to develop an alternative that enables to label tweets more reliably than with the second approach and in greater quantities than with the first approach. Our contributions are as follows:

(1) We propose an annotation pipeline (see Figure 1) that proceeds in two stages: One for annotating the mention of a medication-intake and one for annotating self-experiential side effect mentions.
(2) We employ a semi-supervised active learning method for the problem of discovering side effects that is shown to be competitive with other state-of-the-art active learning methods[a].
(3) We introduce a new method to ensure the representativeness of the training batches.
(4) We describe a workflow for obtaining Twitter data for a given list of drugs and for prefiltering this Twitter data for potential side effects.
(5) For the first time we combine MTurk and active learning with the goal of discovering side effects in Twitter data.

In Section 2 we describe our methods. In Section 3 we explain our experimental settings and report our results on an external dataset and on our own dataset. Finally, we discuss the results in Section 4, touch upon related work from a technical point of view in Section 5, and conclude in Section 7.

[a]The method was previously presented as a poster at the Bayesian Deep Learning Workshop at NeurIPS 2018. `http://bayesiandeeplearning.org/2018/papers/6.pdf`

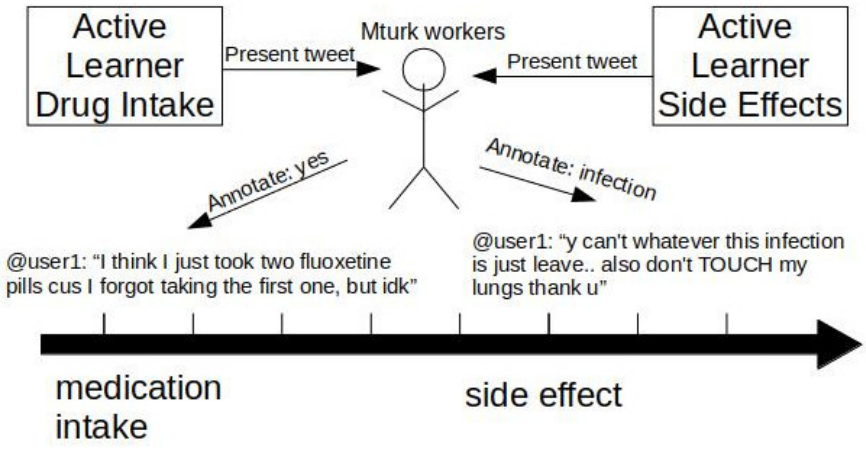

Fig. 1. Workflow.

2. Methods

2.1. *Labeling tweets using MTurk*

This section describes our novel workflow for using MTurk to discover side effects.

The labeling process consists of two stages (see also Figure 1). In the first stage, tweets are labeled according to the drug that was matched. The workers are asked if the author of the tweets really reports taking the drug. In the second stage, the tweets are prefiltered according to the results of the first stage. Only tweets that were posted after a user reported taking a drug are considered.[b] In this stage, the workers are asked if the tweet contains a side effect and have to select it from a drop down list with auto-completion. Both stages employ the active learning framework introduced in Sections 2.3–2.5.

The instructions are shown below the tweet and a side effect can be typed into a text box with auto-completion[c]. Each batch consists of 100 tweets and contains five test tweets that are designed to test annotators. Annotators who label the test tweets incorrectly are excluded from working on batches in the future. Two test tweets are positive, i.e. they contain a side effect or mention a medication intake ("My headache drives me crazy! Off work since 4 days now", "Omg Aleve was my life saver this morning"). This helps to exclude workers who always answer "None". The other three test tweets are negative and help to exclude workers who disregard negated statements such as "I don't have a headache" or other cases where possible side effects are mentioned, but not as a side effect. They also have to check if a tweet is self-experiential. E.g. the tweet "My uncle has a headache" should result in the answer "None" because the author did not experience the effect himself. Each tweet is labeled by three different workers. Workers can choose a side effect out of a list of side effects given by the SIDER database.[6] If no side effect is mentioned in the tweet, workers should type "None". If there is a side effect, but the worker cannot find it in the provided list, there is an optional

[b]To determine this, we use the classifier of the first stage to rank all tweets. We then choose the number of positive tweets according to the proportion in the test set.
[c]jQuery Typeahead 2.10.6 `https://www.npmjs.com/package/jquery-typeahead`

free text field where comments can be entered. We separated 500 of the remaining tweets as a test set. For this test set we required 5 annotations for each tweet and performed majority voting to arrive at the final annotation. Another independent test set was obtained at a later stage.

2.2. *Extracting Twitter Dataset*

We focus on the so-called TG-Gate drugs,[7] a number of drugs for which toxicogenomic data is available that enables to explore possible mechanistic explanations for side effects in the future. The list of TG-Gate drugs was matched to the Drugbank database, leading to a list of 130 drugs. Using the Drugbank database, synonyms, product names and mixture names were extracted. This list was further reduced according to different criteria. We filtered out words such as "Capsule/s", "Tablet/s", "Caplet's", "Tab's", "Pieces", "Srt", "Src", "Imp" and acronyms related to drug behaviour such as "CD", "CR", "LA", "SR" and "TR". Also dosage information and time frames like "7-day" were removed from the drug names. Drug names shorter than 5 characters or longer than 20 characters were removed because they were frequently not relevant drug names anyone would mention in a tweet of 140 characters. Drug names that are the same for multiple drugs were also removed. Additionally, a number of words was removed manually that led to confusion with side effects or other things such as "Heartburn Relief", "Crazy Coconut" or "Leader All Day". We ended up with a list of 1,684 search terms that we will publish along with our code on github. We initially downloaded 99,755 tweets using these search terms. These tweets mentioned 97 out of the 130 drugs we were looking for. Subsequently, we also collected all tweets that were posted up to seven days following one of the initial tweets by the same user. As one does not know the precise dosage or whether it was single dose or long-term use, one week appears as a suitable heuristic choice for a cut-off date. This led to an additional dataset of 4,141,233 tweets.

The data was extracted using the Twitter public API over the course of ca. one month in December and January 2017. Tweets with links or media and retweets were excluded, tweets not written in English and tweets by the user SickAnimalBot were excluded as well. Further processing of the tweets included the removal of stop words and the application of a tweet tagger[8d]. We ignored tweets related to the drug caffeine, because there were disproportionately many tweets about coffee. We ignored tweets with an URL or email address (based on the tagger output) and stemmed the words using the PorterStemmer as implemented in Lucene.

For the bag of words representation of the data, we deleted the punctuation, abbreviations, foreign words, possessive endings, symbols and numbers. Hashtags were replaced with <hashtag>, emoticons with <emoticon>, @mention with <mention>, also based on the tagger output. We then built the vocabulary by choosing the words that occur more than five times and add remaining side effects and synonyms of side effects. This leads to a vocabulary size of 5,601 for the first stage and 4,669 and 4,741 for the active and random run of the second stage, respectively. Duplicates according to the bag of words representation are deleted and tweets with less than four words are removed as well.

[d] `http://www.cs.cmu.edu/~ark/TweetNLP/#parser_down`, model.20120919

2.3. *Semi-supervised Variational Autoencoders*

The proposed method is derived from variational autoencoders, with our own additions for active learning described towards the end of this subsection and in Subsections 2.4 and 2.5. Variational autoencoders are generative models that put a prior distribution on the latent variables (Kingma and Welling[9]). The observed input x is assumed to be generated by a distribution $p_\theta(z)p_\theta(x|z)$ that involves the latent variable z and its prior distribution $p_\theta(z)$. The intractable posterior $p_\theta(z|x)$ is approximated by a variational distribution $q_\phi(z|x)$. The two distributions are parameterized by θ and ϕ that are the parameters of the decoder and the encoder of the autoencoder neural network, respectively.

The semi-supervised VAE[10] optimizes a different objective dependent on whether the label y is observed or not. If the label is observed, the objective is:

$$\log p_\theta(x,y) \geq \mathbb{E}_{p_\phi(z|x,y)}[\log p_\theta(x|y,z) + \log p_\theta(y) + \log p_\theta(z) - \log q_\phi(z|x,y)] = -\mathcal{L}(x,y)$$

In the other case with an unobserved label the objective is:

$$\begin{aligned}&\log p_\theta(x) \geq \mathbb{E}_{p_\phi(y,z|x)}[\log p_\theta(x|y,z) + \log p_\theta(y) + \log p_\theta(z) - \log q_\phi(y,z|x)] \\ &= \sum_y q_\phi(y|x)\left(-\mathcal{L}(x,y)\right) + \mathcal{H}\left(q_\phi(y|x)\right) = -\mathcal{U}(x)\end{aligned}$$

We implement this by adding each unlabeled example twice, once with a positive label and once with a negative label, which enables us to modify the last part of the equation by removing the weight $q_\phi(y|x)$, effectively giving both classes equal weight. This reduces the variance of the loss, especially in the early stages of training, and results in a similar performance:

$$-\mathcal{U}'(x) = -\mathcal{L}(x,y) + \mathcal{H}(q_\phi(y|x))$$

The bound for the labeled part of the dataset is then given by:

$$\mathcal{J}_L = \sum_{(x,y)\sim\tilde{p}_l} \mathcal{L}(x,y), \tag{1}$$

where $\tilde{p}_l$ and $\tilde{p}_u$ refer to the empirical distributions over the labeled and unlabeled subsets of the whole dataset respectively, and the bound for the unlabeled part is given by

$$\mathcal{J}_U = \sum_{(x,y)\sim\tilde{p}_l} \sum_{x\sim\tilde{p}_u} \mathcal{U}'(x).$$

If we want to use this model for classification, we have to add a classification loss to Equation 1, resulting in the extended objective

$$\mathcal{J}_L^\alpha = \mathcal{J}_L + \alpha \cdot \mathbb{E}_{\tilde{p}_l(x,y)}[-\log q_\phi(y|x)],$$

where α is a hyperparameter that controls how much weight is given to the discriminative part of learning. It is set to $\alpha = 0.1 \cdot N$.

In our AL method, we train with all remaining unlabeled data. However, in the beginning of training, this gives a lot of weight to the unlabeled data. Therefore, we train with the labeled data for more iterations in order to achieve a balance between unlabeled and labeled training data. If L is the number of labeled data and U is the number of unlabeled data, we train with the labeled data for $\lceil \frac{U}{L} \rceil$ iterations.

Algorithm 1 Active Learning Algorithm

Input: Unlabeled Dataset D, Initial Labeled Data L
Output: Labeled Instances L, Trained Classifier C

while not converged **do**
  Train classifier: $C \leftarrow Train(L)$
  Compute uncertainty score (Equation 2) for all $d \in D$
  Select 900 most uncertain instances $U \subset D$
  $R = U$
  while $|R| > 100$ **do**
    Train a classifier C to distinguish R and D
    $R \leftarrow R \backslash$instance that belongs to R according to C
  end while
  Obtain labels for selected instances R using MTurk
  $L \leftarrow L \cup R$
  $D \leftarrow D \setminus R$
end while

2.4. *Uncertainty*

We compare two different uncertainty measures for our method. The first is entropy: $H(p(y|x)) = -\mathbb{E}_{p(y|x)}[\log p(y|x)]$ and the second is the Bayesian active learning by disagreement (BALD) uncertainty measure:[11]

$$\mathbb{I}[y, \omega|x, \mathcal{D}_{train}] \approx -\sum_c \left(\frac{1}{T}\sum_t \hat{p}_c^t\right) \log\left(\frac{1}{T}\sum_t \hat{p}_c^t\right) + \frac{1}{T}\sum_{c,t} \hat{p}_c^t \log \hat{p}_c^t, \tag{2}$$

where $\hat{p}_c^t$ is the estimated probability for class c in dropout iteration t and T is the total number of dropout iterations.

2.5. *Representativeness*

Uncertainty strategies are known to request outliers that are not representative of the data as a whole. This is why we also take the representativeness into account. We compare two different strategies. The first strategy requests labels for instances x_i that maximize $Z(x_i) * u_i$, where u_i is the uncertainty and the density $Z(x_i)$ is given as follows:[12]

$$Z(x_i) = \exp\left(\frac{1}{|D|}\sum_{x_h} -\beta(D_{KL}(p(W|x_h)||\lambda p(W|x_i) + (1-\lambda)p(W)))\right),$$

where W is a random variable over the vocabulary, D_{KL} is the Kullback-Leibler divergence, λ is a smoothing parameter and β determines the sharpness of the distance metric.

The second strategy consists of two steps. First, the most uncertain instances are selected. Here we select a number that is larger than what we actually need. In the second step we randomly pick a subset of the uncertain instances that were selected in the first step. This has the effect that we still select uncertain instances, but representativeness is also taken into account through the random sampling step.

As a third strategy we consider a more targeted version of strategy two. Instead of randomly sampling instances, we iteratively remove instances that belong to the uncertainty pool with high confidence (see Algorithm 1, lines 6–9). This means we train a classifier to differentiate the uncertain instances from the whole dataset repeatedly and remove one instance at each step until we arrive at a new training set of instances that are maximally uncertain, but also representative in the sense that the training set cannot be distinguished from the entire pool by a classifier.

3. Results

3.1. *Experimental Setting*

First we evaluated our method on an existing dataset of labeled tweets[13,14] with 6,455 tweets that are classified according to whether or not they mention a side effect of a drug. Approximately 10% of the data are labeled as positive. The vocabulary size is 2,651 after pruning stop words and stemming. For the CNN method we did not employ stemming to make better use of the word vectors.

We compare our method to two different baselines. First, we compare to the Naive Bayes with EM method,[12] which is similar to our method in that it also uses semi-supervised learning. Second, we compare to the deep Bayesian AL method by Siddhant and Lipton[15] as a recent state-of-the-art method based on CNNs. For the first method we use our own reimplementation, whereas for the second method we use the code provided by the authors[e].

The parameters for our method are commonly used default parameters and set as follows: 75 topics, a learning rate of 0.001, a batch size of 50, two layers of hidden neurons with 512 and 256 neurons respectively, we take one sample and we train with early stopping using 10% of the current labeled training set for validation. The architecture is the same as in ProdLDA.[16] We report results averaged over 10 runs of 5-fold cross validation. All methods are trained on an initial batch of 100 random documents and subsequently we add 100 new documents in each AL acquisition step.

3.2. *Comparison with other Active Learning Methods*

In the upper left plot of Figure 2 we compare different AL methods. The Naive Bayes method with EM clearly fails on this Twitter dataset. However, our method is better than the CNN method by Siddhant and Lipton.

The upper right plot of Figure 2 compares the supervised and semi-supervised variant of our method. We can see that the AL variant is better than the non-active variant and the semi-supervised AL variant has an even steeper performance increase in the beginning of training. Thus, we show that both components of our method, the AL component and the semi-supervised component, work well together and improve over the non-active baseline.

[e]We adapted the implementation (`https://github.com/asiddhant/Active-NLP`) to the same training and testing setting that is used for our method. In particular we modified it to not use the test set for selecting instances.

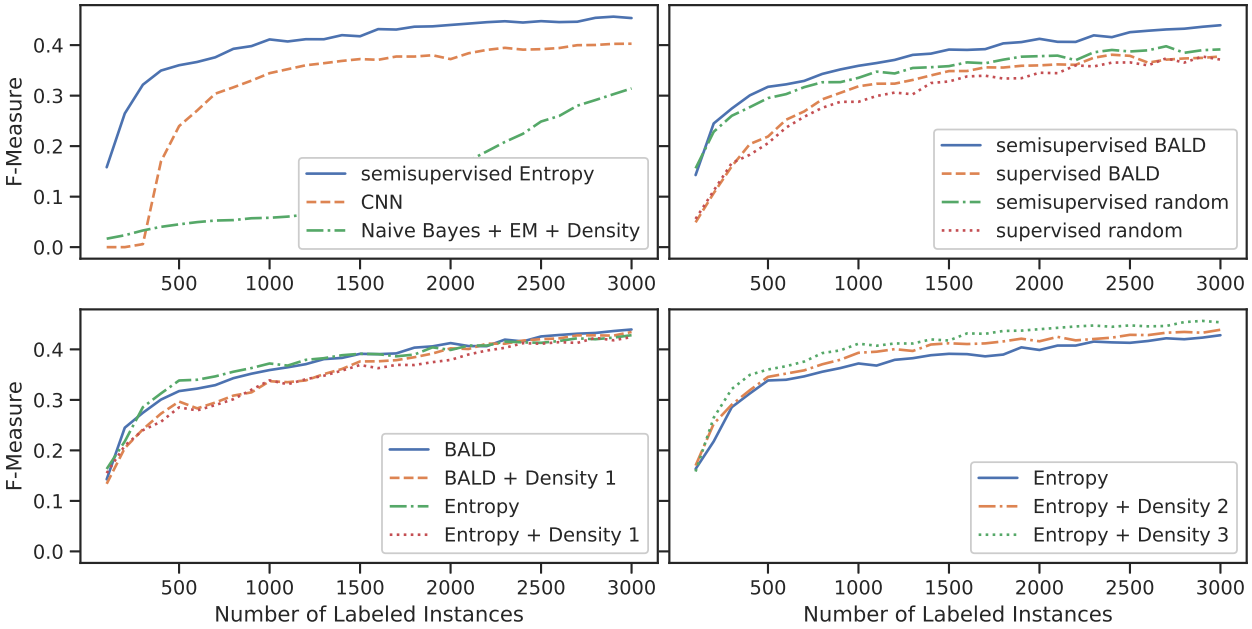

Fig. 2. Plotted is the F-measure against the number of labeled documents averaged over 10 runs of 5-fold cross validation; upper left: different AL methods are compared; upper right: the supervised and semi-supervised variant are compared; lower left: different uncertainty strategies are tested, lower right: different density methods are tested. All results shown are for the 1 stage process.

In the lower left plot of Figure 2 we test different uncertainty strategies. From these results we cannot conclude that one strategy is preferable to another. Entropy and BALD perform more or less on par. Also, the first density strategy[12] does not seem to have a positive effect.

The other density strategies are evaluated in the lower right plot of Figure 2. Here we can see that subsampling a larger pool of uncertain examples according to strategy three as described in Algorithm 1 and Section 2.5 leads to a better performance as compared to strategies one and two. Therefore, we decided to use this strategy with a pool of 900 examples for our labeling experiments on MTurk.

3.3. *Results on Active Learning with MTurk*

To summarize the dataset we obtained through annotation with MTurk, some statistics are shown in Table 2. Overall we collected a dataset of 8,201 labeled tweets divided into annotations for medication intake and in a second stage annotations for self-experiential side effect mentions. In each stage, we did an active learning run and a baseline run where the tweets were randomly selected for annotation.

After stage 1 was completed, we used the medication intake tweets to filter the potential side effect mentions of the second stage. That means, we only suggested potential side effect mentions for annotation where a medication intake was confirmed through the classifier that was trained in the first stage.

Results are shown in Figure 3. We show the F-measure, precision and recall on the test set over the course of one training run. The number of training examples is shown on the

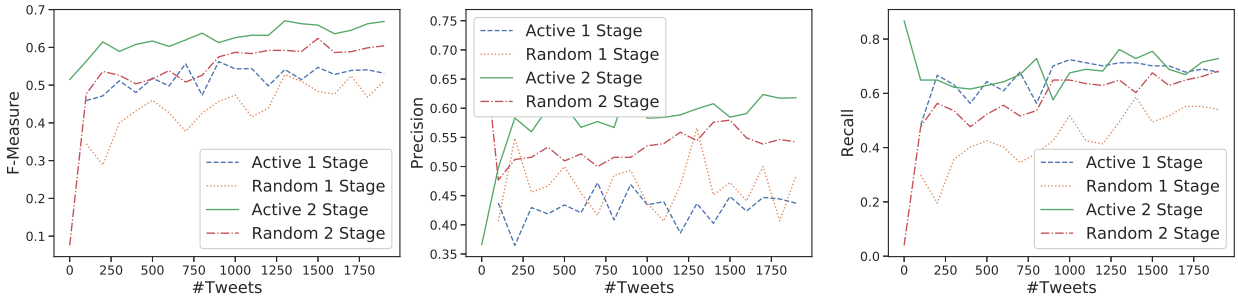

Fig. 3. F-measure, precision and recall on the test set for the MTurk experiments, where 1 stage means, the side effect mention is predicted directly, and 2 stage means the tweets are processed in two stages.

x-axis. We compare the active learning method with the baseline where tweets are selected at random. Additionally, we compare the annotation in just one stage with our proposed two stage annotation pipeline. In the one stage annotation, side effect mentions are annotated directly without first annotating the medication intake tweet.

We can see that the active method in two stages improves the F-measure by almost 15% as compared to the one stage active learning method. Additionally, the active learning procedure improves over the random procedure in both cases. Considering the difficulty of the task and that previous work only achieves an F-measure of up to 0.52 in the detection of adverse drug reaction in tweets,[17] our result of 0.67 is clearly a step in the right direction. Of course this result relies on the prefiltering through the first stage and thus is not directly comparable. However, our proposed solution is promising for potential practical applications. Overall, we found 4,446 different side effect mentions for 45 different drugs.

4. Discussion

A well-known problem of active learning is the deviation of the training set distribution from the original dataset distribution, which may negatively impact generalization performance. It is therefore important to reduce the training set bias that is introduced through active learning. We evaluated two different methods for making the actively selected dataset more representative. The method that subsamples from a larger pool of uncertain examples resulted in a small improvement. This method simply selects examples at random, but more complex schemes for selecting instances are possible and can be investigated in the future.

To give one example, we focus on Naproxen. Table 1 lists the most frequent side effects for Naproxen that were found in our dataset. Naproxen is used for the treatment of pain, menstrual cramps, rheumatoid arthritis, and fever. The most common side effects according to the SIDER database are headache, dispepsia, influenza, pain, and asthenia. Accordingly, these are among the most mentioned effects on Twitter. Additionally, we found frequent side effects on Twitter that are not reported as frequent in SIDER: depression (11 mentions), anxiety (4), sleep related issues (8) as well as various other effects such as skin or muscle problems, showing that it may be possible to see potential side effects on Twitter that were not reported in initial studies.

Table 1. Side effects of Naproxen. Naproxen is used for treatment of pain, menstrual cramps, rheumatoid arthritis and fever. Shown are the most frequent side effects according to the SIDER database. We summarized all side effects related to headache, pain etc. For example, pain includes back pain, chest pain, hand pain, eye pain, ear pain, neuralgia and injection site pain. Influenza includes all influenza-like side effects such as cough or sneezing.

side effect	Freq. SIDER	Freq. Our Data
Headache	15%	19
Dyspepsia	14%	10
Influenza	10%	18
Pain	3-9%	19
Asthenia	1-3%	20

Table 2. Statistics on the tweets labeled using MTurk.

Stage 1–medication intake	
#tweets labeled	4,001
#tweets with medication intake	1,151
#tweets without medication intake	2,850
#approved annotators	210
#excluded annotators	96
#annotations per tweet (train/test)	3/5
Stage 2–self-experiential side effect	
#tweets labeled	4,200
#tweets with a side effect	1,667
#tweets without a side effect	2,523
#approved annotators	138
#excluded annotators	76
#annotations per tweet (train/test)	3/5

We conducted the annotation in two stages. The first stage of annotating medication intake serves the purpose of prefiltering tweets in the second stage. Without prefiltering, our pool of tweets in the second stage was 79,888 tweets. Through removing tweets without an associated medication intake we reduced this to 37,886 and 38,029 for the active learning run and the random baseline, respectively. This filtering leads to the improved precision we observed in our results, since many potential false positives are removed.

5. Limitations

Our approach is currently only applicable on bag-of-words data, meaning that the order of the words is completely disregarded. Considering word order in VAEs is still an active research topic and not straightforward to solve.

The source data is limited as it relies on self-reports. While some users provide information about their location, sex and age, others do not or provide false information. However, this can also be seen as a strength of the approach since users are able to communicate more freely in the perceived anonymity provided by the internet. Nevertheless, the data is necessarily biased, which means that side effects, that are discovered in this way, need to be studied further and be complemented with mechanistic explanations or further evidence.

The results in Table 3 were obtained on a test set that was collected at the same time as the training set. We confirmed the results on an independent test set in subsequent experiments where we find that the two stage process is effective, however, the advantage of active learning as compared to the random baseline is slightly reduced, which could be due to the different data distribution in the new test set.

6. Related Work on Active Learning and Variational Autoencoders

This section discusses related work for the method we employ for semi-supervised active learning, which was presented at the Bayesian Deep Learning Workshop at NeurIPS 2018.

To reduce the cost of labeling, at least two different approaches can be taken. First, semi-supervised training takes advantage of unlabeled data to learn the data distribution and use this implicit information to improve classification. Second, in active learning (AL) the algorithm can choose which documents will be labeled. In this way, the number of labeled examples can be reduced if the chosen examples are informative. Often, uncertainty is used as a measure for how informative a training example is, but representativeness of the whole dataset is also an important factor.

Existing work on deep Bayesian AL is based on Bayesian CNNs,[18] a dropout-based approach, or on the Bayes-by-Backprop (BBB) algorithm.[19] These methods place a prior on the weights of the neural network. Gal *et al.*[18] used Bayesian CNNs for AL on image data. Siddhant and Lipton[15] compared Bayesian CNNs and BBB for text classification, named entity recognition and semantic role labeling. Both of these approaches are purely supervised and cannot take advantage of unlabeled data.

In contrast to BBB, variational autoencoders (VAEs)[9,10] place a prior on the latent variables directly, enabling semi-supervised training to discover latent factors. Recent work on text data explored the use of different priors and network architectures.[20] The neural variational document model (NVDM)[21] is a VAE with a Gaussian prior, whereas ProdLDA[16] uses a Laplace approximation to a Dirichlet prior. In this work we build on recent work on implicit reparameterization gradients[22] to train our network with a Dirichlet prior on the latent variables. The implicit reparameterization gradients are combined with a semi-supervised framework that is evaluated in different settings.

While Gal *et al.*[18] have compared AL to semi-supervised methods and found similar performance, the method we use is the first to combine both, deep Bayesian AL and semi-supervised learning.

7. Conclusion

In this work we show how Amazon Mechanical Turk in conjunction with a state-of-the-art semi-supervised active learning classifier can be used to efficiently extract side effects from twitter data in a process with two stages. Active learning was shown to perform significantly better than the non-active baseline on data labeled on MTurk. Thus, we show for the first time the feasibility of using active learning and crowd sourcing for the task of extracting side effects from twitter data. The data obtained from this study will be used in the next step to search for potentially unknown side effects, to compare the frequency of the occurrence of side effects as compared to reported frequencies in the SIDER database and to make a connection to mechanistic explanations for side effects using the toxicogenomic database TG-GATEs.[7]

Acknowledgements

Funding for this work was provided by Computational Science Mainz (CSM).

References

1. L. J, P. BH and C. PN, Incidence of adverse drug reactions in hospitalized patients: A meta-analysis of prospective studies, *JAMA* **279**, 1200 (1998).

2. F. Laws, C. Scheible and H. Schütze, Active learning with amazon mechanical turk, in *Proceedings of the conference on empirical methods in natural language processing*, 2011.
3. N. Alvaro, Y. Miyao and N. Collier, Twimed: Twitter and pubmed comparable corpus of drugs, diseases, symptoms, and their relations., *JMIR Public Health and Surveillance* **3** (2017).
4. R. Eshleman and R. Singh, Leveraging graph topology and semantic context for pharmacovigilance through twitter-streams, *BMC Bioinformatics* **17**, p. 335 (Oct 2016).
5. A. R. Aronson, Effective mapping of biomedical text to the umls metathesaurus: the metamap program., in *Proceedings of the AMIA Symposium*, 2001.
6. M. Kuhn, I. Letunic, L. J. Jensen and P. Bork, The sider database of drugs and side effects, *Nucleic Acids Research* **44**, D1075 (2016).
7. Y. Igarashi, N. Nakatsu, T. Yamashita, A. Ono, Y. Ohno, T. Urushidani and H. Yamada, Open tg-gates: a large-scale toxicogenomics database, *Nucleic acids research* **43**, D921 (2014).
8. K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan and N. A. Smith, *Part-of-speech tagging for twitter: Annotation, features, and experiments*, tech. rep., Carnegie-Mellon Univ Pittsburgh Pa School of Computer Science (2010).
9. D. P. Kingma and M. Welling, Auto-encoding variational Bayes, in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.
10. D. P. Kingma, S. Mohamed, D. J. Rezende and M. Welling, Semi-supervised learning with deep generative models, in *Advances in Neural Information Processing Systems*, 2014.
11. N. Houlsby, F. Huszár, Z. Ghahramani and M. Lengyel, Bayesian active learning for classification and preference learning, *arXiv preprint arXiv:1112.5745* (2011).
12. A. McCallum and K. Nigam, Employing EM and pool-based active learning for text classification, in *ICML*, 1998.
13. A. Sarker and G. Gonzalez, Portable automatic text classification for adverse drug reaction detection via multi-corpus training, *Journal of biomedical informatics* **53**, 196 (2015).
14. R. Ginn, P. Pimpalkhute, A. Nikfarjam and A. Patki, Mining twitter for adverse drug reaction mentions: A corpus and classification benchmark, in *Proceedings of the Fourth Workshop on Building and Evaluating Resources for Health and Biomedical Text Processing (BioTxtM2014)*, 2014.
15. A. Siddhant and Z. C. Lipton, Deep bayesian active learning for natural language processing: Results of a large-scale empirical study, in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.
16. A. Srivastava and C. Sutton, Autoencoding variational inference for topic models, in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.
17. A. Magge, A. Sarker, A. Nikfarjam and G. Gonzalez-Hernandez, Comment on: "Deep learning for pharmacovigilance: recurrent neural network architectures for labeling adverse drug reactions in Twitter posts", *Journal of the American Medical Informatics Association* **26**, 577 (04 2019).
18. Y. Gal, R. Islam and Z. Ghahramani, Deep Bayesian active learning with image data, in *ICML*, eds. D. Precup and Y. W. Teh, Proceedings of Machine Learning Research, Vol. 70 (PMLR, International Convention Centre, Sydney, Australia, 06–11 Aug 2017).
19. C. Blundell, J. Cornebise, K. Kavukcuoglu and D. Wierstra, Weight uncertainty in neural networks, in *ICML*, ICML'15 (JMLR.org, 2015).
20. S. Burkhardt and S. Kramer, Decoupling sparsity and smoothness in the dirichlet variational autoencoder topic model, *Journal of Machine Learning Research* **20**, 1 (2019).
21. Y. Miao, L. Yu and P. Blunsom, Neural variational inference for text processing, in *ICML*, eds. M. F. Balcan and K. Q. Weinberger, Proceedings of Machine Learning Research, Vol. 48 (PMLR, New York, New York, USA, 20–22 Jun 2016).
22. M. Figurnov, S. Mohamed and A. Mnih, Implicit reparameterization gradients, in *Advances in Neural Information Processing Systems 31*, eds. S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi and R. Garnett (Curran Associates, Inc., 2018) pp. 439–450.