# View Distillation with Unlabeled Data for Extracting Adverse Drug Effects from User-Generated Data

**Payam Karisani**
Emory University
pkarisa@emory.edu

**Jinho D. Choi**
Emory University
jinho.choi@emory.edu

**Li Xiong**
Emory University
lxiong@emory.edu

## Abstract

We present an algorithm based on multi-layer transformers for identifying Adverse Drug Reactions (ADR) in social media data. Our model relies on the properties of the problem and the characteristics of contextual word embeddings to extract two views from documents. Then a classifier is trained on each view to label a set of unlabeled documents to be used as an initializer for a new classifier in the other view. Finally, the initialized classifier in each view is further trained using the initial training examples. We evaluated our model in the largest publicly available ADR dataset. The experiments testify that our model significantly outperforms the transformer-based models pretrained on domain-specific data.

## 1 Introduction

Social media has made substantial amount of data available for various applications in the financial, educational, and health domains. Among these, the applications in healthcare have a particular importance. Although previous studies have demonstrated that the self-reported online social data is subject to various biases (Olteanu et al., 2018), this data has enabled many applications in the health domain, including tracking the spread of influenza (Aramaki et al., 2011), detecting the reports of the novel coronavirus (Karisani and Karisani, 2020), and identifying various illness reports (Karisani and Agichtein, 2018).

One of the well-studied areas in online public health monitoring is the extraction of adverse drug reactions (ADR) from social media data. ADRs are the unintended effects of drugs for prevention, diagnosis, or treatment. The researchers in Duh et al. (2016) reported that consumers, on average, report the negative effect of drugs on social media 11 months earlier than other platforms. This highlights the importance of this task. Another team of researchers in Golder et al. (2015) reviewed more than 50 studies and reported that the prevalence of ADRs across multiple platforms ranges between 0.2% and 8.0%, which justifies the difficulty of this task. In fact, despite the long history of this task in the research community (Yates and Goharian, 2013), for various reasons, the performance of the state-of-the-art models is still unsatisfactory. Social media documents are typically short and their language is informal (Karisani et al., 2015). Additionally, the imbalanced class distributions in ADR task has exacerbated the problem.

In this study we propose a novel model for extracting ADRs from Twitter data. Our model which we call View Distillation (VID) relies on the existence of two views in the tweets that mention drug names. We use unlabeled data to transfer the knowledge from the classifier in each view to the classifier in the other view. Additionally, we use a finetuning technique to mitigate the impact of noisy pseudo-labels after the initialization (Karisani and Karisani, 2021). As straightforward as it is to implement, our model achieves the state-of-the-art performance in the largest publicly available ADR dataset, i.e., SMM4H dataset. Our contributions are as follows: 1) We propose a novel algorithm to transfer knowledge across models in multi-view settings, 2) We propose a new technique to efficiently exploit unlabeled data in the supervised ADR task, 3) We evaluate our model in the largest publicly available ADR dataset, and show that it yields an additive improvement to the common practice of language model pretraining in this task. To our knowledge, our work is the first study that reports such an achievement. Next, we provide a brief overview of the related studies.

## 2 Related Work

Researchers have extensively explored the applications of ML and NLP models in extracting ADRs from user-generated data. Perhaps one of the early reports in this regard is published in Yates and



*Proceedings of the Sixth Social Media Mining for Health Workshop 2021*, pages 7–12

Goharian (2013), where the authors utilize the related lexicons and extraction patterns to identify ADRs in user reviews. With the surge of neural networks in text processing, subsequently, the traditional models were aggregated with these techniques to achieve better generalization (Tutubalina and Nikolenko, 2017). The recent methods for extracting ADRs entirely rely on neural network models, particularly on multi-layer transformers (Vaswani et al., 2017).

In the shared task of SMM4H 2019 (Weissenbacher and Gonzalez-Hernandez, 2019), the top performing run was BERT model (Devlin et al., 2019) pretrained on drug related tweets. Remarkably, one year later in the shared task of SMM4H 2020 (Gonzalez-Hernandez et al., 2020), again a variant of pretrained BERT achieved the best performance (Liu et al., 2019). Here, we propose an algorithm to improve on pretrained BERT in this task. Our model relies on multi-view learning and exploits unlabeled data. To our knowledge, our model is the first approach that improves on the domain-specific pretrained BERT.

## 3 Proposed Method

Our model for extracting the reports of adverse drug effects rely on the properties of contextual neural word embeddings. Previous research on Word Sense Disambiguation (WSD) (Scarlini et al., 2020) has demonstrated that contextual word embeddings can effectively encode the context in which words are used. Although the representations of the words in a sentence are assumed to be distinct, they still possess shared characteristics. This is justified by the observation that the techniques such as self-attention (Vaswani et al., 2017), which a category of contextual word embeddings employ (Devlin et al., 2019), rely on the interconnected relations between word representations.

This property is particularly appealing when documents are short, therefore, word representations, if are adjusted accordingly, can be exploited to extract multiple representations for a single document. In fact, previous studies have demonstrated that word contexts can be used to process short documents, e.g., see the models proposed in Liao and Grishman (2011) and Karisani et al. (2020) for event extraction using hand-crafted features and contextual word embeddings respectively. Therefore, we use the word representations of drug mentions in user postings as the secondary view along the document representations of user postings in our model.

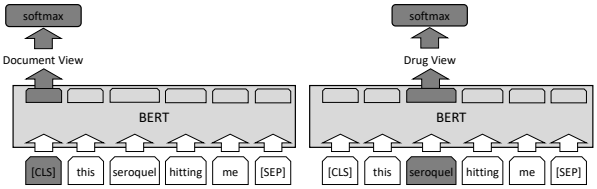


Figure 1: The illustration of the document and drug views in our model. We have used BERT as an encoder. See Devlin et al. (2019) for the format of input tokens.

As a concrete example, from the hypothetical tweet *"this seroquel hitting me"*, we extract one representation from the entire document and another representation from the drug name[1] Seroquel. In continue, we call these two views the document and drug views. Figure 1 illustrates these two views using BERT (Devlin et al., 2019) as an encoder.

Given the two views we can either concatenate the two sets of features and train a classifier on the resulting feature vector or use a co-training framework as described in Karisani et al. (2020). However, the former is not exploiting the abundant amount of unlabeled data, and the latter is resource intensive, because it is iterative, and also it has shown to be effective only in semi-supervised settings where there are only a few hundred training examples available. Therefore, below we propose an approach to effectively use the two views along the available unlabeled data in a supervised setting.

In the first step, we assume the classifier in each view is a student model and train this classifier using the pseudo-labels generated by the counterpart classifier. Since the labeled documents are already annotated, we carry out this step using the unlabeled documents. More concretely, let $L$ and $U$ be the sets of labeled and unlabeled user postings respectively. Moreover, let $L_d$ and $L_g$ be the sets of representations extracted from the document and drug views of the training examples in the set $L$; and let $U_d$ and $U_g$ be the document and drug representations of the training examples in the set $U$. To carry out this step, we train a classifier $C_d$ on the representations in $L_d$ and probabilistically, with temperature $T$ in the softmax layer, label the representations in $U_d$. Then we use the association between the representations in $U_d$ and $U_g$ to construct a pseudo-labeled dataset of $U_g$. This dataset along its set of probabilistic pseudo-labels is used in a distillation technique (Hinton et al., 2015) to train a classifier called $\widehat{C_g}$. Correspondingly, we

[1] We assume every user posting contains only one drug name, in cases that there are multiple names we can use the first occurrence.

use the set $L_g$ to train a classifier $C_g$, then label the set $U_g$ and use the association between the data points in $U_g$ and $U_d$ to construct a pseudo-labeled dataset in the document view to train the classifier $\widehat{C_d}$.

The procedure above results in two classifiers $\widehat{C_d}$ and $\widehat{C_g}$. The classifier in each view is *initialized* by the knowledge transferred from the other view. However, the pseudo-labels that are used to train each classifier can be noisy. Thus, in order to reduce the negative impact of this noise, in the next step, we use the training examples in the sets $L_d$ and $L_g$ to further finetune these two classifiers respectively. To finetune $\widehat{C_d}$ we use the objective function below:

$$\mathcal{L}_d = \frac{1}{|L_d|} \sum_{v \in L_d} (1-\lambda) J(\widehat{C_d}(v), y_v) + \lambda J(\widehat{C_d}(v), C_d(v)), \quad (1)$$

where $J$ is the cross-entropy loss, $y_v$ is the ground-truth label of the training example $v$, and $\lambda$ is a hyper-parameter to govern the impact of the two terms in the summation. The first term in the summation, is the regular cross-entropy between the output of $\widehat{C_d}$ and the ground-truth labels. The second term is the cross-entropy between the outputs of $\widehat{C_d}$ and $C_d$. We use the output of $C_d$ as a regularizer to train $\widehat{C_d}$ in order to increase the entropy of this classifier for the prediction phase. Previous studies have shown that penalizing low entropy predictions increases generalization (Pereyra et al., 2017). We argue that this is particularly important in the ADR task, where the data is highly imbalanced. Note that, even though $C_d$ is trained on the training examples in $L_d$, the output of this classifier for the training examples is not sparse–particularly for the examples with uncommon characteristics. Thus, we use these soft-labels[2] along the ground-truth labels to train $\widehat{C_d}$. Respectively, we use the objective function below to finetune $\widehat{C_g}$:

$$\mathcal{L}_g = \frac{1}{|L_g|} \sum_{v \in L_g} (1-\lambda) J(\widehat{C_g}(v), y_v) + \lambda J(\widehat{C_g}(v), C_g(v)), \quad (2)$$

where the notation is similar to that of Equation 1. Here, we again use the output of $C_g$ as a regularizer to train $\widehat{C_g}$. In the evaluation phase, to label the unseen examples, we take the average of the outputs of the two classifiers $\widehat{C_d}$ and $\widehat{C_g}$.

Algorithm 1 illustrates our model (VID) in Structured English. On Lines 8 and 9 we derive the document and drug representations from the sets $L$ and $U$. On Lines 10 and 11 we use the labeled training examples in the two views to train $C_d$ and $C_g$. On Lines 12-14 we train and finetune $\widehat{C_g}$, and on Lines 15-17 we train and finetune $\widehat{C_d}$. Finally, we return $\widehat{C_d}$ and $\widehat{C_g}$. In the next section, we describe our experimental setup.

**Algorithm 1** Overview of VID

1: **procedure** VID
2: **Given:**
3: $L$ : Set of labeled documents
4: $U$ : Set of unlabeled documents
5: **Return:**
6: Two classifiers $\widehat{C_d}$ and $\widehat{C_g}$
7: **Execute:**
8: Derive two sets of representations $L_d$ and $L_g$ from $L$
9: Derive two sets of representations $U_d$ and $U_g$ from $U$
10: Use $L_d$ to train classifier $C_d$
11: Use $L_g$ to train classifier $C_g$
12: Use $C_d$ to probabilistically label $U_d$
13: Transfer labels of $U_d$ to $U_g$ and use them to train $\widehat{C_g}$
14: Finetune $\widehat{C_g}$ using Equation 2
15: Use $C_g$ to probabilistically label $U_g$
16: Transfer labels of $U_g$ to $U_d$ and use them to train $\widehat{C_d}$
17: Finetune $\widehat{C_d}$ using Equation 1
18: **Return** $\widehat{C_d}$ and $\widehat{C_g}$

## 4 Experimental Setup

We evaluated our model in the largest publicly available ADR dataset, i.e., the SMM4H dataset. This dataset consists of 30,174 tweets. The training set in this dataset consists of 25,616 tweets of which 9.2% are positive. The labels of the test set are not publicly available. The evaluation in the dataset must be done via the CodaLab website. We compare our model with two sets of baselines: 1) a set of baselines that we implemented, 2) the set of baselines that are available on the CodaLab website[3].

Our own baseline models are: **BERT**, the base variant of the pretrained BERT model (Devlin et al., 2019), as published by Google. **BERT-D**, a domain-specific pretrained BERT model. This model is similar to the previous baseline, however, it is further pretrained on 800K unlabeled drug-related tweets that we collected from Twitter. We pretrained this model for 6 epochs using the next sentence prediction and the masked language model tasks. **BERT-D-BL**, a bi-directional LSTM model. In this model we used BERT-D followed by a bi-directional LSTM network (Hochreiter and Schmidhuber, 1997).

[2] Again, we use temperature $T$ in the softmax layer to train using the soft-labels.

[3] Available at: https://competitions.codalab.org/SMM4H. The 2020 edition of the shared task is not online anymore. Therefore, for a fair comparison with the baselines, we do not use RoBERTa in our model, and instead use pre-trained BERT model.

| Type | Method | F1 | Precision | Recall |
|---|---|---|---|---|
| Our Impl. | BERT | 0.57 | 0.669 | 0.50 |
| | BERT-D | 0.62 | 0.736 | 0.54 |
| | BERT-D-BL | 0.61 | **0.749** | 0.52 |
| CodaLab | Sarthak | 0.65 | 0.661 | 0.65 |
| | leebean337 | 0.67 | 0.600 | **0.76** |
| | aab213 | 0.67 | 0.608 | 0.75 |
| | VID | **0.70** | 0.678 | 0.72 |

Table 1: F1, Precision, and Recall of our model (VID) in comparison with the baselines.

We also compare our model with all the baselines available on the CodaLab webpage. These baselines include published and unpublished models. They also cover models that purely rely on machine learning models and those that heavily employ medical resources; see Weissenbacher and Gonzalez-Hernandez (2019) for the summary of a subset of these models.

We used the Pytorch implementation of BERT (Wolf et al., 2019). we used two instances of BERT-D as the classifiers in our model–see Figure 1. Please note that using domain-specific pretrained BERT in our framework makes any improvement very difficult, because the improvement in the performance should be additive. We used the training set of the dataset to tune for our two hyper-parameters $T$ and $\lambda$. The optimal values of these two hyper-parameters are 2 and 0.5 respectively. We trained all the models for 5 epochs[4]. During the tuning, we observed that the finetuning stage in our model requires much fewer training steps, therefore, we finetuned for only 1 epoch. In our model, we used the same set of unlabeled tweets that we used to pretrain BERT-D. This verifies that, indeed, our model extracts new information that cannot be extracted using the regular language model pretraining. As required by SMM4H we tuned for F1 measure. In the next section, we report the F1, Precision, and Recall metrics.

## 5 Results and Analysis

Table 1 reports the performance of our model in comparison with the baseline models–only the top three CodaLab baselines are listed here. We see that our model significantly outperforms all the baseline models. We also observe that the performances of our implemented baseline models are lower than that of the CodaLab models. This difference is mainly due to the gap between the size of the unlabeled sets for the language model pretraining in the experiments–ours is 800K, but the top CodaLab model used a corpus of 1.5M examples. This suggests that our model can potentially achieve a better performance if there is a larger unlabeled corpus available.

[4]We used 20% of the training set for validation, and observed that the models overfit if we train more than 5 epochs.

| Method | F1 | Precision | Recall |
|---|---|---|---|
| *Document-View* | 0.62 | 0.736 | 0.54 |
| *Drug-View* | 0.63 | 0.706 | 0.570 |
| *Combined-View* | 0.63 | 0.745 | 0.543 |
| *VID* | 0.70 | 0.678 | 0.72 |

Table 2: F1, Precision, and Recall of VID in comparison to the performance of the classifiers trained on the document, drug, and combined views.

| Method | F1 | Precision | Recall |
|---|---|---|---|
| *P-Doc-F-Doc* | 0.69 | 0.658 | 0.71 |
| *P-Drug-F-Drug* | 0.68 | 0.681 | 0.68 |
| *P-Doc-F-Drug* | 0.70 | 0.674 | 0.72 |
| *P-Drug-F-Doc* | 0.69 | 0.655 | 0.72 |
| *VID* | 0.70 | 0.678 | 0.72 |

Table 3: Performance of VID in comparison to the performance of the classifiers pretrained on the document or drug pseudo-labels (indicated by P-{•}) and finetuned on the document or drug training examples (indicated by F-{•}).

Table 2 reports the performance of VID in comparison to the classifiers trained on the document and drug representations. We also concatenated the two representations and trained a classifier on the resulting feature vector, denoted by *Combined-View*. We see that our model substantially outperforms all three models. Table 3 compares our model with the classifiers with different pretraining and finetuning resources. Again, we see that VID is comparable to the best of these models. We also observe 2 percent absolute improvement by comparing *P-Drug-F-Drug* and *P-Doc-F-Drug*, which signifies the efficacy of View Distillation.

In summary, we evaluated our model in the largest publicly available ADR dataset and compared with the state-of-the-art baseline models that use domain specific language model pretraining. We showed that our model outperforms these models, even though it uses a smaller unlabeled corpus. We also carried out a set of experiments and demonstrated the efficacy of our proposed techniques.

## 6 Conclusions

In this study we proposed a novel model for extracting adverse drug effects from user generated content. Our model relies on unlabeled data and a novel technique called view distillation. We evaluated our model in the largest publicly available ADR dataset, and showed that it outperforms the existing BERT-based models.

## References


Eiji Aramaki, Sachiko Maskawa, and Mizuki Morita. 2011. Twitter catches the flu: Detecting influenza epidemics using twitter. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1568–1576, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc of the 2019 NAACL*, pages 4171–4186.

Mei Sheng Duh, Pierre Cremieux, Marc Van Audenrode, Francis Vekeman, Paul Karner, Haimin Zhang, and Paul Greenberg. 2016. Can social media data lead to earlier detection of drug-related adverse events? *Pharmacoepidemiology and Drug Safety*, 25(12):1425–1433.

Su Golder, Gill Norman, and Yoon K Loke. 2015. Systematic review on the prevalence, frequency and comparative value of adverse events data in social media. *British Journal of Clinical Pharmacology*, 80(4):878–888.

Graciela Gonzalez-Hernandez, Ari Z. Klein, Ivan Flores, Davy Weissenbacher, Arjun Magge, Karen O'Connor, Abeed Sarker, Anne-Lyse Minard, Elena Tutubalina, Zulfat Miftahutdinov, and Ilseyar Alimova, editors. 2020. *Proceedings of the Fifth Social Media Mining for Health Applications Workshop & Shared Task*. Association for Computational Linguistics, Barcelona, Spain (Online).

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Negin Karisani and Payam Karisani. 2020. Mining coronavirus (covid-19) posts in social media. *arXiv preprint arXiv:2004.06778*.

Payam Karisani and Eugene Agichtein. 2018. Did you just have a heart attack?: Towards robust detection of personal health mentions in social media. In *Proc of the 2018 WWW*, pages 137–146.

Payam Karisani, Eugene Agichtein, and Joyce Ho. 2020. Domain-guided task decomposition with self-training for detecting personal events in social media. In *Proceedings of The Web Conference 2020*, WWW '20, page 2411–2420, New York, NY, USA. Association for Computing Machinery.

Payam Karisani and Negin Karisani. 2021. Semi-supervised text classification via self-pretraining. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, page 40–48. Association for Computing Machinery.

Payam Karisani, Farhad Oroumchian, and Maseud Rahgozar. 2015. Tweet expansion method for filtering task in twitter. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, pages 55–64.

Shasha Liao and Ralph Grishman. 2011. Using prediction from sentential scope to build a pseudo co-testing learner for event extraction. In *Proc of 5th IJCNLP*, pages 714–722.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Alexandra Olteanu, Emre Kiciman, and Carlos Castillo. 2018. A critical review of online social data: Biases, methodological pitfalls, and ethical boundaries. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, pages 785–786, New York, NY, USA. ACM.

Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. 2017. Regularizing neural networks by penalizing confident output distributions. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020. Sensembert: Context-enhanced sense embeddings for multilingual word sense disambiguation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8758–8765. AAAI Press.

Elena Tutubalina and Sergey Nikolenko. 2017. Combination of deep recurrent neural networks and conditional random fields for extracting adverse drug reactions from user reviews. *Journal of healthcare engineering*, 2017.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Davy Weissenbacher and Graciela Gonzalez-Hernandez, editors. 2019. *Proceedings of the Fourth Social Media Mining for Health Applications (#SMM4H) Workshop & Shared Task*. Association for Computational Linguistics, Florence, Italy.

Thomas Wolf, Lysandre Debut, and et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Andrew Yates and Nazli Goharian. 2013. Adrtrace: Detecting expected and unexpected adverse drug reactions from user reviews on social media sites. In *Advances in Information Retrieval*, pages 816–819, Berlin, Heidelberg. Springer Berlin Heidelberg.