A Two Phase Deep Learning Model for Identifying Discrimination from Tweets

Shuhan Yuan
Tongji University
Shanghai, China
4e66@tongji.edu.cn

Xintao Wu
University of Arkansas
Fayetteville, AR, USA
xintaowu@uark.edu

Yang Xiang
Tongji University
Shanghai, China
shxiyangyang@tongji.edu.cn

ABSTRACT

Discrimination discovery is the data mining problem of unveiling discriminatory practices by analyzing a dataset of historical decision records. In this paper, we focus on discovering discrimination from tweets using deep learning models. One challenge here is that it is difficult to obtain a large well-labeled dataset required by the training of deep learning models for the purpose of discrimination analysis. We develop a two-phase deep learning model to address this challenge. Our model first learns text representations based on weakly-labeled tweets (containing some specific hashtags), then trains the classifier on a small set of well-labeled training data. Experimental results show that: (1) the proposed method can be successfully used for discrimination identification; (2) pre-training text representations, which utilizes weakly-labeled tweets, can significantly improve the accuracy of discrimination detection.

Keywords
deep learning; discrimination analysis; two phase learning

1. INTRODUCTION

Discrimination generally refers to an unjustified distinction of individuals based on gender, race, or religion, and often occurs when the group (e.g., female) is treated less favorably than others. Discrimination discovery and prevention from historical databases has been an active research area recently. In this paper, we are focused on a related but different problem, i.e., how to identify discriminatory tweets. For example, if an individual publishes a tweet saying “Want to learn photography or how to use photo shop? It’s men’s lifestyle interest. Not for girls!”, obviously this tweet contains discrimination against female. Identifying discrimination from text is an important task in user-generated content (UGC) mining as discrimination has increasingly become a hotspot of social attention nowadays.

Recent work in natural language processing has shown that deep learning models could learn meaningful representations (or features) of text and train to classify text on top of text representations with high accuracy in applications like text classification and sentiment analysis. In this paper, we examine the use of deep learning models for discrimination analysis of tweets. However, existing deep learning models require large amounts of training data and it is difficult to obtain such a large well-labeled training dataset (each tweet is clearly marked with discrimination or non-discrimination by domain users) because labeling manually a large number of tweets is time-consuming.

We develop a two-phase deep learning model to detect discrimination from tweets. In the first phase, the model focuses on learning semantic representations of tweets using the large amount of weakly-labeled tweets. In Twitter, users often add hashtags, which mark keywords or topics, in their tweets. We consider tweets containing hashtags like “#sexism”, “#racism” are weakly-labeled discrimination tweets and those tweets likely contain discrimination information. One example is “Why are female cabinet members suspect but male ones are not? #bias #sexism”. However, not all tweets containing such hashtags can be considered as discrimination. For example, the tweet “#sexism is an important research in behavior research” is not discriminatory. In general, the tweets that are weakly-labeled by discrimination-related hashtags are likely to be discriminatory than those without discrimination-related hashtags. Hence we train our model to learn the good text representations based on the similarity between the weakly-labeled tweets and well-labeled tweets. In the second phase, we use the representations of tweets trained from the first phase as inputs to train the logistic regression classifier and fine-tune the whole model using the small set of well-labeled tweets.

2. THE MODEL

In this section, we describe the two-phase deep learning model to identify discrimination tweets.

2.1 Phase One

In the first phase, we first model tweets representations based on semantic composition ideas [4]. Semantic composition aims to understand the text by composing the meaning of each word through a composition function. In our work, we use the Long Shot-Term Memory (LSTM) [3] recurrent neural network as the composition function to model the features of tweets. LSTM is able to model a tweet by sequentially processing each word and mapping a tweet to a low dimensional representation vector. LSTM has various variations. In our work, we adopt a widely used LSTM model...
but without peephole connections. In order to learn a
tweet representation, the model first maps each word \( w_i \) in a
tweet into a \( d \)-dimensional real vector \( x_w \in \mathbb{R}^d \), also called
word embeddings \([1]\). For a tweet with \( n \) words, a sequence
of word embeddings \( x = (x_1, x_2, \ldots, x_n) \) are passed into the
LSTM one by one to compute the sequential hidden feature
vectors \( h = (h_1, h_2, \ldots, h_n) \). Then, the model combines the
hidden vectors by mean operation \( r = \text{mean}(h_1, h_2, \ldots, h_n) \) to
get one vector \( r \) as representation of the tweet.

We further need to model the feature of discrimination and
non-discrimination category. In order to build the dis-
crimination features, we consider all the discrimination tweets
as a document and use the features of each tweet as input to
compose the discrimination features of each category. Given
a set of discrimination tweets \( T^+ = \{t_1^+, t_2^+, \ldots, t_m^+\} \), after com-
puting the representations of tweets \( R^+ = \{r_1^+, r_2^+, \ldots, r_m^+\} \), we
composite the representations of discrimination by Equation 1.

\[
Q^+ = \frac{1}{m} \sum_{i=1}^{m} r_i^+.
\]

To build the representations of non-discrimination \( Q^- \in \mathbb{R}^{d} \),
the framework follows the same procedure.

The objective of our model is to let the representations of
weakly-labeled tweets close to the representations of similar
category and far away to the representations of their
opposite category. For example, if a tweet contains hash-
tag “#sexism”, we want the representation of this tweet
close to the representation of discrimination and far from
the representation of non-discrimination. Our model uses
cosine function \( \text{sim}(r, Q^+) \) (\( \text{sim}(r, Q^-) \)) to measure the sim-
ilarity between a weakly-labeled tweet representation \( r \) and
representation of discrimination (non-discrimination) cate-
gory. If \( r \) is a weakly-labeled discrimination tweet, we set
\( \delta = \text{sim}(r, Q^+) - \text{sim}(r, Q^-) \). If \( r \) is weakly-labeled as non-
discrimination tweet, we set \( \delta = \text{sim}(r, Q^-) - \text{sim}(r, Q^+) \). The
loss function is \( L(\delta) = \log(1 + \exp(-\gamma\delta)) \), where \( \gamma \) is a scal-
ing factor. To train the model, we use the back-propagation
algorithm by Adadelta \([5]\) to update the parameters of the
LSTM model.

### 2.2 Phase Two

In the second phase, we aim to learn the logistic regression
classifier to identify discrimination. After the pre-training,
the LSTM model which contains word embeddings to the
semantic representations of tweets is already well-trained.
We stack the logistic regression layer on the LSTM layer
and feed the tweets representations as inputs to logistic reg-
ression classifier. We use the well-labeled small dataset
as training dataset in this phase. The model is to predict
whether a tweet contains discrimination \( \hat{y} \). The logistic reg-
ression function is:

\[
P(\hat{y}|r, U_b, b_l) = \frac{1}{1 + e^{-(U_b r + b_l)}}.
\]

where \( r \) is the representation of a tweet, and \( U_b, b_l \) are the
parameters of logistic regression. We use negative log likel-
hood as the loss function to train the classifier and fine-tune
the whole architecture.

### 3. EXPERIMENTS

We crawled tweets online and labeled 300 discrimination
tweets and 300 non-discrimination tweets as the well-labeled
data set. Meanwhile, we treated 2000 tweets with “#sexism”
or “#racism” as weakly-labeled discrimination data and 2000
tweets with “#news” as weakly-labeled non-discrimination
data. To evaluate the performance, we split the well-labeled
dataset into training data and test data with different sizes
and use 5-fold cross validation to evaluate the classification
performance. We compare our model with several baselines,
which include the deep learning without pre-training the
tweets representations, SVM, and Naive Bayes classifiers.
We use 1-gram and 2-gram as features of SVM and Naive
Bayes classifiers. The prediction results are shown in Table
1. We observe our deep learning model significantly outper-
forms SVM and Naive Bayes classifiers and the pre-training
further improves the accuracy.

### 4. CONCLUSIONS AND FUTURE WORK

We presented a two-phase deep learning model for dis-
crimination analysis of tweets. Our model first learns text rep-
resentations based on weakly-labeled tweets (containing some
specific hashtags), then trains the classifier on a small set
of well-labeled training data. The preliminary experiments
showed that pre-training text representations by weakly-
labeled tweets could improve the accuracy of discrimination
detection. Meanwhile, our model can be easily extended
to other applications that are restricted by lack of a large
amount of training data. In the future, we plan to extend our
method to identify more fine-grained discrimination text.

### 5. ACKNOWLEDGMENTS

This work was supported in part by The 973 Program
of China (2014CB340404) and China Scholarship Council.
This research was conducted while the first author visited
University of Arkansas.

### 6. REFERENCES

[1] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A
neural probabilistic language model. *Journal of


distributional models of semantics. *Cognitive Science*,


<table>
<thead>
<tr>
<th>Methods</th>
<th>Number of training data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>240</td>
</tr>
<tr>
<td>Our model</td>
<td>0.887</td>
</tr>
<tr>
<td>Without pre-training</td>
<td>0.870</td>
</tr>
<tr>
<td>SVM (1-gram)</td>
<td>0.724</td>
</tr>
<tr>
<td>SVM (2-gram)</td>
<td>0.736</td>
</tr>
<tr>
<td>Naive Bayes (1-gram)</td>
<td>0.827</td>
</tr>
<tr>
<td>Naive Bayes (2-gram)</td>
<td>0.852</td>
</tr>
</tbody>
</table>