Classifying Hate Speech Using a Two-Layer Model

Yiwen Tang & Nicole Dalzell

To cite this article: Yiwen Tang & Nicole Dalzell (2019) Classifying Hate Speech Using a Two-Layer Model, Statistics and Public Policy, 6:1, 80-86, DOI: 10.1080/2330443X.2019.1660285

To link to this article: https://doi.org/10.1080/2330443X.2019.1660285

© 2019 The Author(s). Published with license by Taylor and Francis Group, LLC

View supplementary material

Accepted author version posted online: 26 Aug 2019. Published online: 01 Oct 2019.

Submit your article to this journal

Article views: 722

View related articles

View Crossmark data
Classifying Hate Speech Using a Two-Layer Model

Yiwen Tang and Nicole Dalzell

Department of Mathematics and Statistics, Wake Forest University, Winston-Salem, NC

ABSTRACT

Social media and other online sites are being increasingly scrutinized as platforms for cyberbullying and hate speech. Many machine learning algorithms, such as support vector machines, have been adopted to create classification tools to identify and potentially filter patterns of negative speech. While effective for prediction, these methodologies yield models that are difficult to interpret. In addition, many studies focus on classifying comments as either negative or neutral, rather than further separating negative comments into subcategories. To address both of these concerns, we introduce a two-stage model for classifying text. With this model, we illustrate the use of internal lexicons, collections of words generated from a pre-classified training dataset of comments that are specific to several subcategories of negative comments. In the first stage, a machine learning algorithm classifies each comment as negative or neutral, or more generally target or nontarget. The second stage of model building leverages the internal lexicons (called L2CLs) to create features specific to each subcategory. These features, along with others, are then used in a random forest model to classify the comments into the subcategories of interest. We demonstrate our approach using two sets of data. Supplementary materials for this article are available online.

1. Introduction

In February 2019, YouTube disabled the comment feature on many videos featuring young children, terminated hundreds of channels featuring minors (Feiner 2019; Shaban 2019), and “remov[ed] hundreds of millions of comments for violating” (YouTube 2019). These actions were in response to a report released by a video blogger indicating that the comment feature of YouTube was being used to identify and disseminate material of interest to child predators and child pornographers (Feiner 2019). In addition to disabling and removing comments, YouTube has since “been working on an even more effective classifier, that will identify and remove predatory comments” (YouTube 2019). This recent instance is only one example of the importance of developing methods to classify and flag content distributed through social media.

The use of social media sites as a platform for communication is a topic of continued research (Smith et al. 2008; O’Keeffe and Clarke-Pearson 2011; Bertot, Jaeger, and Hansen 2012; Hassan et al. 2017). While online comments have exhibited some positive impacts on users, there is also the potential for harm (Smith et al. 2008; O’Keeffe and Clarke-Pearson 2011); the extreme sentiments appearing on YouTube videos featuring young children are only one such example. However, identifying extreme comments on social media is a complex task. Individuals expressing extreme sentiments through social media comments can strive to bypass detection systems. For instance, an internet user can replace a letter with a digit (i.e., replacing “o” with “0”), delete vowels (i.e., replacing “fish” with “fsh”), or replace a word with another word with similar pronunciation (i.e., replacing “son” with “sun”). These changes can help comments evade detection systems, making it difficult to identify negative comments. The process of identification is further complicated by the reality that online communities do not always practice rules of grammar or the general structure of a sentence.

In the literature, there are two major approaches used to classify comments: machine learning approaches and lexicon-based approaches. Authors who adopt the machine learning approach may apply complex machine learning algorithms, such as convolutional neural networks and support vector machines, to classify comments based on features derived from a dataset of comments (Pang, Lee, and Vaithyanathan 2002; Hailong, Wenyan, and Bo 2014; Hassan et al. 2017; Singla, Randhawa, and Jain 2017; Chen and Zhang 2018). These models can have high prediction accuracy, however, the final models tend to be difficult to interpret due to the complexity of the algorithms. This means that while the models may be able to predict the negativity of a comment, it may not yield insight into what defines these classifications.

Insights into what defines a classification of a comment is provided more naturally by lexicon-based classification approaches (Pang, Lee, and Vaithyanathan 2002; Hailong, Wenyan, and Bo 2014; Hassan et al. 2017; Singla, Randhawa, and Jain 2017). These methods define categories based on a dictionary, a collection of pre-sorted words referring to a particular topic. Some examples of dictionaries include WordNet (Nasim, Rajput, and Haider 2017) or Happiness...
Index (Gonçalves et al. 2013). Comments are then compared to these dictionaries using a variety of metrics, and these metrics are used to assign comments to categories. Due to their reliance on a dictionary, the performance of lexicon-based classification depends greatly on the availability of a high quality dictionary. In an instance like the YouTube comments, a dictionary appropriate to identifying toxic comments may not have been available.

In this article, we illustrate a method designed to merge the benefits of the machine learning and lexicon-based classification approaches. Specifically, we leverage a collection of human labeled training data in which comments have been identified as belonging to one of J subcategories. From this training data, we create \( J \) lexicons, each one identifying key words within its respective subcategory. We refer to these \( J \) lexicons as L2CLs. The L2CLs are then used to engineer features specific to each of the \( J \) subcategories. These are then used in our classification process.

The classification process itself proceeds in two layers. In the first layer, we apply a boosted tree to classify comments as Negative or Neutral. More generally, this layer is used to classify data into two primary categories: the target category and the nontarget category. In the second layer, we further sort the Negative (i.e., target) comments into subcategories using a random forest model. Each random forest uses features derived from the L2CLs, as well as other data based features, to classify comments into the appropriate subcategories.

The remainder of the article proceeds as follows. In Section 2, we present our two-layer methodology. Section 2.1 details the construction of the L2CLs. In Sections 3 and 4, we conduct two analyses using different datasets to evaluate our method. We conclude with a discussion of results and future work in Section 5.

2. Methods

In this section, we introduce our two-layer classification model, as well as the creation of the L2CLs. In Section 2.1, we detail the process of creating lexicons based on pre-labeled training data. In Section 2.2, we discuss the first layer of the classification model, the purpose of which is to distinguish Negative and Neutral comments. The second layer of our model is introduced in Section 2.3. This layer classifies Negative comments into secondary categories.

We note that our model assumes some data cleaning is performed before classification is attempted. This cleaning process will be specific to each dataset and application of interest. In our applications, the comments are converted to all lowercase letters and duplicate words are removed.

2.1. Building the Lexicons

There is extensive literature on the use of external lexicons for sentiment analysis. For instance, research conducted by Zhang et al. (2011) utilizes several opinion lexicons containing words revealing a certain emotional orientation to classify sentences as positive or negative. The words “good” and “great,” for instance, belong to the lexicon of positive polarity. If these words appear in a sentence, the model predicts a high likelihood that the sentence is positive (Zhang et al. 2011).

While useful when they are available, it is not always reasonable to rely on external lexicons to construct classifiers. Due to some realities of online comments, such as the presence of typos, external lexicons may not include all the derivatives or possible spellings of a word. In addition, a lexicon of a specific subcategory of interest, for instance, Hate Speech versus Directed Threats, may not be available. Furthermore, external lexicons may not exist in any form for certain classification tasks.

In our work, we explore the use of lexicons derived from training data to assist in classification tasks when external lexicons are not available. Our motivating dataset is a large collection of comments from Wikipedia’s Talk Page Edits (Wikipedia 2018; Dixon, Wulczyn, and Thain 2019). The motivating task is to sort these data into \( J = 6 \) pre-identified subcategories indicating how “toxic” each comment is. With such specific subcategories, an external lexicon is not available to classify comments. However, a set of pre-labeled training data is available. In our approach, we leverage this training data to construct \( J \) Layer 2 Classification Lexicons (L2CLs), one for each subcategory present in the training data.

The general process of building an L2CL for subcategory \( j = 1, \ldots, J \) begins by identifying and storing the unique words in each comment in the pre-labeled training dataset. The unique words may comprise of words that are not spelled correctly; this reality is discussed in Section 3.1. At this stage, each unique spelling is considered a unique word. Once comments have been reduced to their unique words, the process of building an L2CL continues by dividing the training data into their pre-labeled subcategories. Within each subcategory \( j \), we pool together the \( m(j) \) unique words \( v_{1j}, v_{2j}, \ldots, v_{m(j)} \) from the comments classified to that subcategory. We denote this set of words as \( \text{Pool}_j = \{v_{1j}, v_{2j}, \ldots, v_{m(j)}\} \). If a word \( v \) is in \( \text{Pool}_j \), then \( v \) appears in at least one comment in subcategory \( j \).

The next step is to obtain a frequency for each word in \( \text{Pool}_j \). Here we define the frequency of \( v \) in \( \text{Pool}_j \) as the number of comments in the training data assigned to subcategory \( j \) that contain word \( v \). We then refine the set \( \text{Pool}_j \) by merging derivative words. For instance, if the words “eat” and “eats” are both in \( \text{Pool}_j \), we do not want to treat them as two distinct elements. Instead, we merge them by their core (“eat”) and add their frequencies together. We refer to this process as core merging.

Our final step is to collect the \( K \) core-merged words in the set \( \text{Pool}_j \) that have the highest frequency; denote these words as \( w_{1j}, \ldots, w_{Kj} \), where each \( w_{ij} \in \text{Pool}_j \). We call this set of \( C \) core-merged words \( \text{L2CL}_j = \{w_{1j}, \ldots, w_{Kj}\} \), where \( \text{L2CL}_j \subset \text{Pool}_j \).

In our applications, we choose to use \( K = 100 \), yielding lexicons that captured most words that occurred multiple times within a subcategory. However, the choice of the number of core-merged words to include in an L2CL, as well as the number of L2CLs constructed, can be adapted to suit a given application. The L2CL lexicons will be used for feature engineering during the second layer of our classification model, so the only restriction is that there must be \( J \) L2CLs constructed, where \( J \) is the number of subcategories. Sensitivity analysis to the number of core words chosen for the L2CL is a subject of future work.
2.2. Classification Layer 1

With the \( J \) L2CLs created, we turn our attention to building a model to classify the comments. The first layer in our model, Layer 1, divides the data into two primary categories. For our motivating dataset, the comments are divided into Negative and Neutral. In this case, Neutral comments are not of interest for the analysis; the focus is on further classifying Negative comments. By applying this first round of classification, users may ignore the Neutral comments and focus solely on comments identified as Negative. We note that this stage can be adapted based on the number of broad categories of interest in a given dataset; users are not restricted to the use of a binary classifier in Layer 1.

There are numerous statistical methods and machine learning algorithms available to perform classification of text data. Because interpretability is not the focus of Layer 1, it is reasonable to use a complex algorithm as the Layer 1 classifier. Machine learning techniques generally rely upon data features to perform classification. These features can be learned by the algorithm or specified by the user. For our applications, we developed 15 features to train the Layer 1 Classifier; these features are described in Appendix B (supplementary materials). However, users may select features that are appropriate to a given application.

In our examples, the model used for the Layer 1 classifier is a boosted tree model. However, this technique can also be adapted to suit realities of other datasets. To select our Layer 1 approach, we compared the performance of several techniques, including random forest and neural network models; for a complete list of the techniques we considered, see Table 9 in Appendix D (supplementary materials). We repeated 5-fold cross-validation three times to estimate the prediction accuracy of each of algorithms under consideration. The performance of XGBoost tree (Chen and Guestrin 2016) and monotonemultilayerperception neural network are roughly equivalent, and we selected the XGBoost tree in our applications as the Layer 1 classifier. A short description of XGBoost tree is included in Appendix C (supplementary materials).

2.3. Classification Layer 2

Once the Layer 1 classifier has sorted the comments into target and nontarget primary categories, our task becomes to further divide comments into more refined subcategories. In the case of the Wikipedia dataset, this means classifying comments designated as Negative in Layer 1 into one of \( J = 6 \) subcategories. The Layer 2 classifier is a random forest model whose features include the 15 features designed for the Level 1 classifier as well as four additional features derived from the \( J \) L2CLs. We fit a random forest for each subcategory \( j = 1, \ldots, J \) using a binary outcome (whether a comment is in subcategory \( j \) or not). We perform this classification in such a way that a comment can be assigned to more than one subcategory. For instance, a Negative comment may be classified in both the subcategory Toxic and the subcategory Obscene. If desired, however, the Layer 2 classifier can be adapted to allow only unique subcategory classifications. This can be done by running a single random forest whose outcome variable is a classifier with \( J \) levels.

In Sections 2.3.1–2.3.4, we detail the process of creating the L2CL-based features that will be used in the Layer 2 classifier.

2.3.1. L2CL-Based Features: The Subcategory Score

The first L2CL-based feature is based on what we call the Subcategory Score (SCS). This score is meant to reflect how often a particular word occurs in a given subcategory \( j \) relative to other words in L2CL\(_j\). For each word \( w_{ij} \) in L2CL\(_j\), we compute the number of comments in subcategory \( j \) containing word \( w_{ij} \), that is, the frequency of word \( w_{ij} \) in L2CL\(_j\). We call this count \( f_{ij} \). We then sum these frequencies across all words \( w_{ij} \) in L2CL\(_j\) to obtain a total frequency of words in L2CL\(_j\); we refer to this value \( F_j = \sum_{w_{ij} \in \text{L2CL}_j} f_{ij} \) as the subcategory total count. Then, for each word \( w_{ij} \), we create a SCS \( s_{ij} \) as

\[
s_{ij} = \frac{f_{ij}}{F_j}.
\]

The SCS of a word \( w_{ij} \) serves as an indicator of the importance of \( w_{ij} \) in subcategory \( j \). However, the score does not reflect the prevalence of word \( w_{ij} \) in the L2CLs of other subcategories. A word can appear in an L2CL for more than one \( j \), and we assume that the number of L2CLs a word belongs to influences this word’s importance as a tool for classification. Accordingly, once we have created the SCS, we design two additional scores for the purpose of reflecting the number of L2CLs a word belongs to. We call these the P1 and P2 scores.

2.3.2. L2CL-Based Features: P1 and P2 Scores

Let \( L_{ij} \) be the number of L2CLs in which word \( w_{ij} \) in L2CL\(_j\) appears. For instance, if the word "edit" appears in five of six L2CLs, \( L_{ij} = 5 \). To obtain our second L2CL-based score (the P2 score), we divide \( s_{ij} \) by \( L_{ij} \), yielding

\[
P2_{ij} = \frac{s_{ij}}{L_{ij}}.
\]

The P2 score incorporates a penalty if a word appears in more than one L2CL. The logic behind this option is that if a word appears in, for example, five L2CLs, it is likely that the word is not iconic enough to distinguish the subcategories. Words that appear only in one subcategory are not penalized.

Our third L2CL-based feature is motivated by our Wikipedia data. In these data, there are two large subcategories (Toxic and Obscene) which are essentially “generally negative” categories of comments. Such categories may not have many specific terms that distinguish them, in contrast to subcategories like Identity-Hate. However, the presence of words contained in multiple L2CLs, that is, “generally negative” terms, might be indicative that a comment belongs to these larger, more general subcategories. To explore this, we create our third L2CL-based score (the P1 score). We multiply \( s_{ij} \) by \( L_{ij} \) to obtain

\[
P1_{ij} = s_{ij} \times L_{ij}.
\]

Performance of these scores in our classification applications is discussed in Section 3.2.

2.3.3. L2CL-Based Features: P3 Score

It is also logical that the relative size of the subcategories is important to both the construction of the L2CL and the classification process. To incorporate the size of the subcategories into our feature engineering, we also introduce a fourth L2CL-based
score, called the P3 score. This score is motivated by realities in the Wikipedia data. As illustrated in Table 1, in our pre-labeled training data for the Wikipedia comments, 94% of the negative comments are labeled in the subcategory Toxic and 52% of the negative comments are classified in the subcategory Obscene.

There is therefore significant overlap in comments appearing in these subcategories. This means that if a word is in the L2CL for Obscene comments, this word is also likely to appear in the L2CL of Toxic comments. This overlap reduces the ability of a word to distinguish the subcategories. To account for this, we create the P3 score, which penalizes $s_j$ if $w_{ij}$ belongs to the L2CL of a dominant subcategory.

Let $n_{target}$ denote the number of comments identified as Negative by the Layer 1 classifier. Let $n_j$ denote the number of comments in the pre-labeled training data that are classified in subcategory $j$. Let $S_j$ denote the set of L2CLs $w_{ij}$ belongs to. For example, if $w_{ij} \in L2CL_1$ and $L2CL_3$, then $S = \{1, 3\}$. The P3 score for L2CL $j$ is then calculated as

$$P3_{ij} = (s_{ij} \prod_{s \in S[-j]} \left(1 - \frac{n_j}{n_{target}}\right))^\frac{1}{|S|-1},$$

(4)

where $S[-j]$ denotes the set including all elements of $S$ except $j$.

### 2.3.4. L2CL-Based Features: Using the Scores

For every subcategory $j$, the four scores (the SCS and P1–P3 scores) are used to generate four features for each comment $c = 1, \ldots, n_{target}$. The four features are called the Subcategory Feature (SCF), P1 Feature, P2 Feature, and P3 Feature.

Suppose we are constructing the feature for comment $c$ for subcategory $j$. Let $\alpha_{ij} = (\alpha_{ij}^1, \ldots, \alpha_{ij}^{100})$ be a binary vector of length 100 indicating if each word $w_{ij}$ in L2CL$_j$ is in comment $c$. For instance, if comment $c$ contains the second word in L2CL$_j$, $\alpha_{ij}^2 = 1$; otherwise, $\alpha_{ij}^2 = 0$. To create the SCF for comment $c$ for subcategory $j$, we compute,

$$SCF_{ij} = \sum_{i=1}^{K} s_{ij} \times \alpha_{ij}.$$ 

(5)

In other words, the SCF is the sum of all $s_{ij}$ such that word $w_{ij}$ is in comment $c$. The other three final scores are computed similarly; the process is detailed in Appendix A.1 (supplementary materials).

### 2.3.5. Layer 2 Classifier

The four L2CL features, along with the 15 features used in Layer 1, are used in Layer 2 to categorize comments into subcategories. The classification is performed independently for each subcategory using a random forest model; a short description of Random Forest model can be found in Appendix C (supplementary materials). Random forests are attractive at this stage because they offer a balance of interpretability and predictive accuracy.

By fitting a separate random forest model for each subcategory, we allow a comment to be placed in more than one subcategory. If unique classification is desired, one random forest model is sufficient.

### 3. Application 1: Wikipedia

We use two datasets to illustrate our classification method. The first is a collection of 159,971 comments from Wikipedia (2018). Each comment in the dataset has been pre-labeled into six subcategories of negative comments: Toxic, Severe Toxic, Obscene, Threat, Insult, and Identity-Hate (Hate).

The counts and percentages of these subcategories are displayed in Table 1. The labels are assigned by human raters and focus on the toxicity level of each comment. Each row in the original dataset contains the text of a comment as well as six binary indicators for whether or not this comment falls into each of the six subcategories. The comments range in length from 5 to 5000 characters and contain between 1 and 1403 unique words. Further details about the characteristics of the comments are displayed in Appendix Table 7 (supplementary materials).

Comments may be classified into more than one subcategory, so there is some overlap in the subcategories. For our work, we identify a comment as a Negative comment if it is classified into at least one of these six subcategories; all other comments are classified as Neutral. The counts and percentages of Negative and Neutral comments are displayed in Appendix Table 6 (supplementary materials).

We construct a training set from 15,225 randomly selected negative comments and 15,225 random selected neutral comments. These comments will be treated as the pre-labeled training data to create the L2CL, as well as to train the model. We also construct a test dataset which is the combination of the remaining 1000 negative comments as well as 1000 neutral comments chosen randomly. We built our training set and test set in this manner to utilize all of the negative comments while keeping a balance between negative and neutral comments.

We note that in the original dataset, around 90% of the comments are neutral. For purposes of testing and training, we chose to work with a small, more balanced dataset for practical considerations. However, the proposed methodology does not require such a balance between the classifications; sensitivity analysis for this balance is a subject for future work.

#### 3.1. Data Considerations

This dataset presents several challenges. Since the data are based on online comments, many standard rules of grammar do not apply. Moreover, some comments are composed of only one or two words repeated multiple times. For instance, one comment is the single word “PIG” repeated 1250 times. For our approach, we handle this repetition by reducing each comment to its unique words. Additionally, we create a feature that reflects the number of unique words versus the number of words in a comment.

---

1. The data are available at https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data
A second challenge for the Wikipedia dataset is spelling errors. When one types comments online, typos and mistakes can be made unintentionally. However, deliberate typos are also made by some internet users as a tool to circumvent the supervision of filters. These realities necessitate pre-processing before classification is attempted. Along with the data cleaning, the construction of the L2CLs allows common typos to be treated as words in and of themselves, meaning these words can be used to distinguish subcategories.

Another consideration for this dataset is that comments in the training data have been labelled by humans. This means that the ratings may reflect the background, knowledge, culture, etc., of the individuals involved in the rating process. On the one hand, this means our method creates an L2CL which can be personalized to suit different rating systems, which may be chosen to reflect datasets of interest. However, if the data do not provide information on the individuals or the rating system that pre-classified the training comments, there may be concerns about the accuracy of the classifications. Different people may have distinct backgrounds, beliefs or other personal biases that can influence the classification of a comment. In other words, a comment that sounds offensive to one group might not bestow the same impression upon another group. For instance, in the original dataset, there are two comments which are identical in every way, but were posted twice and hence are recorded as different comments. However, only one of these two comments is labeled as an Identity-Hate comment.

These realities suggest that the pre-labelling may impact the performance of our L2CL-based features. Because of this, we use a second dataset composed of Twitter comments in which the pre-labelling is less subjective than the Wikipedia comments. This application is discussed in Section 4.

3.2. Results

We use three different model fitting procedures on the Wikipedia data, each designed to assess a different component of our model. In Model 1, we use only the Layer 2 classifier to sort comments without first subsetting into Negative and Neutral; in other words, we exclude the Layer 1 classifier. In Model 2, we use both the Layer 1 and Layer 2 classifiers, but do not use the L2CL-based features in Layer 2. In Model 3, we use the full model, including both Layer 1 and Layer 2 classifiers and incorporating the L2CL-based features in Layer 2.

In Model 1, we applied six random forest binary classification models (one to each subcategory) to the entire training dataset. Once the model was trained, we applied the model to classify the 1000 comments in the test dataset. We note that our test dataset is pre-classified, so we are able to compare the results of our model with human-generated subcategories. The results indicate an overall accuracy of 22.6%, meaning 22.6% of the predicted classifications agree with the pre-classified labels. This suggests that, with the chosen features, the random forest model alone is not sensitive enough to effectively classify the comments.

In Model 2, we re-introduce the Layer 1 classifier, but exclude the L2CL-based features from the Layer 2 classifier. For our Layer 1 classifier, we apply an XGBoost classification tree to sort the training comments into Negative or Neutral. For Layer 2, we then apply random forest models to further classify the comments identified as Negative by the Layer 1 classifier. The prediction accuracy on our test data raises to 53.95%, increasing the accuracy by a factor of 2.3. This suggests that adding the first layer of classification can result in improved prediction accuracy.

For Model 3, we add the use of the L2CL-based features to Layer 2. Adding these features results in a modest increase in overall prediction accuracy, increasing to 55.6%. To examine the importance of the L2CL-based features in the classification, we consider relative importance plots. These plots indicate the decrease in test MSE that results from a permutation test conducted on each variable, and then scaled relative to the decrease in MSE on other variables. Figure 1 shows a relative importance plot for the Obscene subcategory. Relative to the other predictors, all four L2CL-based features have a strong impact on

![Figure 1](image-url)
the prediction accuracy of the model within this subcategory. The prediction accuracy for each subcategory are displayed in Table 2. The importance plots for the other subcategories can be found in Appendix Figures 3–7 (supplementary materials).

A key advantage of our lexicon-based features is the interpretability. With the inclusion of L2CL-based features, we are able to conduct cross-subcategory comparisons and learn which words contribute significantly to the classification of a certain subcategory. By comparing the six L2CLs, we found that if a comment reveals some sense of negativity or profanity, it is probably classified as a Toxic comment. Compared to other L2CLs, the Severe Toxic L2CL tends to have more disrespectful terms targeting a person. For instance, the word “idiot” appears in 289 Severe Toxic comments. Obscene comments tend to contain more vulgar terms relevant to body parts and disabilities. Comments that are classified as Threat incorporate many aggressive verbs, such as “kill”; this term appears in 26.8% of the threat comments. In contrast to the L2CL of the category Severe Toxic, the L2CL of Identity-Hate consists of more abusive terms targeting a group of people with the same gender, racial, religious, or sexual orientation identities. We did not find any iconic words, however, in the L2CL of Insult. The reason is that the L2CL of Insult is very similar to that for subcategory Severe Toxic. In fact, out of 1498 Severe Toxic comments, 1290 of them are also classified as insults. This reinforces the notion discussed in Section 3.1 that the quality and distinctness of the subcategory classifications can impact the performance of our model. In Section 4, we explore this in more detail with our second data application.

4. Application 2: Twitter Comments

The second dataset we use contains 14,485 Twitter comments relevant to travel on a commercial airline.2 The data were collected in February 2015, and contributors were asked to first divide the comments into three portions: Positive, Negative, and Neutral. Then, the Negative tweets were further classified by reasons for their negative status, such as Customer Service Issues or Late Flight. There are 10 subcategories in total, but some of these appear infrequently in the training data, as shown in Appendix Table 15 (supplementary materials). For our application, we consider only the Positive tweets as well as comments in the three most popular subcategories: Customer Service Issues, Lost Luggage, and Late Flight.

Similar to the test and training creation for the Wikipedia dataset, we construct a random subset of 4400 tweets. This subset consists of 2200 positive random comments and 2200 negative comments. From this, we sample a training dataset with 2000 positive comments along with 2000 negative comments, and a test dataset is the remaining 200 positive comments and 200 negative comments. The distribution of the subcategories is the same in the training and test data and is displayed in Table 3. Since the tweets in this dataset are collected online, they present similar challenges to those we have discussed in detail in Section 3.1.

We recall that the distinction between the subclassification of the Negative comments in the Twitter and Wikipedia datasets is that the Twitter comments are assigned to exactly one subcategory, while the Wikipedia comments can be assigned to multiple subcategories. Furthermore, the Twitter subcategories are less subjective. Examining the performance of our model on the Twitter data therefore allows us to compare the performance of our model under these conditions. We note, however, that our Layer 2 classifier was not constrained to assigning a unique subcategory, as we were interested in seeing the ability of our model to distinguish the subcategories without this restriction.

| Table 2. | Model 3 results: the accuracy of the six random forest classification models. |
|----------|-------------------------------------------------|
| Subcategory | Toxic | Severe Toxic | Obscene | Threat | Insult | Hate |
| Prediction accuracy | 0.760 | 0.952 | 0.856 | 0.987 | 0.824 | 0.952 |

| Table 3. | The distribution of the categories in the Twitter dataset. |
|----------|-------------------------------------------------|
| Classification | Customer service issues | Late flight | Lost luggage | Positive |
| Count | 1000 | 800 | 200 | 2000 |
| Proportion | 0.25 | 0.20 | 0.05 | 0.50 |

| Table 4. | The performances of the Layer 2 classifier on the test dataset for Model 2 (M2) and Model 3 (M3). |
|----------|-------------------------------------------------|
| Model | Accuracy | Difference |
| M2: Late Flight | 0.78 | |
| M3: Late Flight (with L2CL Features) | 0.84 | 0.06 |
| M2: Customer Service Issues | 0.72 | |
| M3: Customer Service Issues (with L2CL Features) | 0.81 | 0.09 |
| M2: Lost Luggage | 0.95 | |
| M3: Lost Luggage (with L2CL Features) | 0.96 | 0.01 |

| Table 5. | The distributions of the accuracies of the random forest models on the training Twitter dataset. |
|----------|-------------------------------------------------|
| Model: Training Data | Min | Q1 | Q2 | Q3 | Max |
| M2: Late Flight | 0.56 | 0.59 | 0.60 | 0.62 | 0.64 |
| M3: Late Flight (with L2CL Features) | 0.80 | 0.81 | 0.83 | 0.84 | 0.86 |
| M2: Customer Service Issues | 0.52 | 0.55 | 0.56 | 0.57 | 0.60 |
| M3: Customer Service I (with L2CL Features) | 0.74 | 0.75 | 0.77 | 0.79 | 0.80 |
| M2: Lost Luggage | 0.89 | 0.90 | 0.90 | 0.90 | 0.90 |
| M3: Lost Luggage (with L2CL Features) | 0.94 | 0.95 | 0.96 | 0.96 | 0.97 |

NOTE: We conduct 5-fold cross-validation for 3 times to see the general performance of the models. The five number summaries below indicate the results across these repetitions.

---

2The data are available at https://www.kaggle.com/crowdflower/twitter-airline-sentiment
for the two most populated subcategories. For the Lost Luggage subcategory, which accounts for only 5% of the data, the improvement is negligible. The overall prediction accuracy of our two-layer model without the L2CL-based features is 58.25%, whereas that of the 2-layer model containing the L2CL-based features is 66.75%. This suggests the L2CL-based features have the ability to boost the prediction accuracy of the classification models for this dataset.

On the training data, the improvement in prediction accuracy obtained by including the L2CL-based features is even more pronounced. As displayed in Table 5, for each subcategory the minimal prediction accuracy of the RF model with L2CL-based features exceeds the best prediction accuracy of the RF model without the L2CL-based features.

As noted with the Wikipedia dataset, the L2CL-based features are identified as important in the Layer 2 classifier; the relevant importance plots are shown in Appendix Figures 8–10 (supplementary materials).

5. Discussion and Future Work

In this article, we developed a two-layer classification model for text comments, specifically focusing on situations where an external lexicon is not available. The approach illustrates the value of lexicon based features in a text classification problem and highlights a method for constructing a lexicon using pre-labelled training data. As shown in the applications, the four lexicon-based features have strong predictive power and add to the interpretability of the classification solution.

The two datasets we choose allow the comparison of model performance when comments are restricted to a single subcategory or allowed to belong to multiple subcategories. In our applications, our model performs better under the former condition. It is also worth noting, however, that the subcategories in the Twitter data are much more clearly defined than the subcategories in the Wikipedia comments. In fact, it is difficult to clearly define the boundaries in the Wikipedia data between, for instance, the subcategory Severe Toxic and the subcategory Insult. For the more distinct subcategories Identity Hate or Threat in the Wikipedia data, the classification created by our model identifies iconic words which can easily distinguish themselves from comments in other categories. The improvement in model performance suggests that the distinction in the subcategories yields to more effective classification, as is true with classification techniques in general when categories of interest are well defined. This suggests the importance of examining the training data prior to the construction of L2CLs or attempting classification.

Since we only build lexicons of unigrams in our studies, potential future work will include building interpretable features based on bigrams or even trigrams. Furthermore, exploring the impact of misspelled words on the L2CL, as well as applying various techniques to reduce such typos prior to the L2CL construction, are additional avenues for future research.

Supplementary Materials

The supplementary material contains links to the data sets used in Sections 3 and 4, as well as the necessary R code. Contained also are Appendices with additional figures and tables summarizing characteristics of the data and results.

Funding

The authors gratefully acknowledge funding from the Wake Forest Summer Research Fellowship and the Starr Travel Grant, both offered by the URECA Center at Wake Forest University.

ORCID

Yiwen Tang https://orcid.org/0000-0001-8182-6223
Nicole Dalzell https://orcid.org/0000-0003-4211-1916

References

Bertot, J. C., Jaeger, P. T., and Hansen, D. (2012), “The Impact of Polices on Government Social Media Usage: Issues, Challenges, and Recommendations,” Government Information Quarterly, 29, 3040. [80]

Chen, T., and Guestrin, C. (2016), “Xgboost: A Scalable Tree Boosting System,” arXiv no. 1603.02754. [82]

Chen, Y., and Zhang, Z. (2018), “Research on Text Sentiment Analysis Based on CNNs and SVM,” in 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA). [80]

Dixon, L., Wulczyn, E., and Thain, N. (2019), “Wikipedia Detox,” available at https://meta.wikimedia.org/wiki/Research:Detox. [81]

Feiner, L. (2019), ”YouTube is Disabling Comments on Videos With Minors After Advertisers Back Out Over Abuse,” available at https://www.cnbc.com/2019/02/28/youtube-is-disabling-comments-on-videos-with-minors.html. [80]

Gonçalves, P., Araújo, M., Benevenuto, F., and Cha, M. (2013), “Comparing and Combining Sentiment Analysis Methods,” in Proceedings of the First ACM Conference on Online Social Networks, COSN ’13, New York, NY, USA: ACM, pp. 27–38. [81]

Hailong, Z., Wenyang, G., and Bo, J. (2014), “Machine Learning and Lexicon Based Methods for Sentiment Classification: A Survey,” in 2014 11th Web Information System and Application Conference. [80]

Hassan, A. U., Hussain, J., Hussain, M., Sadiq, M., and Lee, S. (2017), “Sentiment Analysis of Social Networking Sites (SNS) Data Using Machine Learning Approach for the Measurement of Depression,” in 2017 International Conference on Information and Communication Technology Convergence (ICTC). [80]

Nasim, Z., Rajput, Q., and Haider, S. (2017), “Sentiment Analysis of Student Feedback Using Machine Learning and Lexicon Based Approaches,” in 2017 International Conference on Research and Innovation in Information Systems (ICRIIS). [80]

O’Keeffe, G. S., and Clarke-Pearson, K. (2011), “The Impact of Social Media on Children, Adolescents, and Families,” Pediatrics, 127, 800–804. [80]

Pang, B., Lee, L., and Vaitheeswaran, S. (2002), “Thumbs Up? Sentiment Classification Using Machine Learning Techniques,” in Proceedings of EMNLP, pp. 79–86. [80]

Shaban, H. (2019), “YouTube Axes Tens of Millions of Comments in Crackdown on Child Sexual Exploitation.” [80]

Singla, Z., Randhawa, S., and Jain, S. (2017), “Sentiment Analysis of Customer Product Reviews Using Machine Learning,” in 2017 International Conference on Intelligengt Computing and Control (I2C2). [80]

Smith, P. K., Mahdavi, J., Carvalho, M., Fisher, S., Russell, S., and Tippet, N. (2008), “Cyberbullying: Its Nature and Impact in Secondary School Pupils,” Journal of Child Psychology and Psychiatry, 49, 376–385. [80]

Wikipedia (2018), “Wikipedia Talk Labels: Personal Attacks [data file],” available at https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/data. [81,83]

YouTube (2019), “More Updates On Our Actions Related to the Safety of Minors on YouTube.” [80]

Zhang, L., Ghosh, R., Dekhil, M., Hsu, M., and Liu, B. (2011), “Combining Lexicon-Based and Learning-Based Methods for Twitter Sentiment Analysis.” [81]