Don't Take This Personally: Sentiment Analysis for Identification of "*Subtweeting*" on Twitter

Noah Segal-Gould   Adviser: Sven Anderson

Experimental Humanities Concentration and Computer Science Program, Bard College, May 2018

Introduction

Twitter is a news and social networking service to which users send text posts called **tweets**. The OED defines "subtweet" as a **tweet** "...that refers to a particular user without directly mentioning them, typically as a form of furtive mockery or criticism."

Figure 1: A Subtweet and a Reply Which Acts as an Accusation

- This project monitors the content of **replies** to acquire a dataset of known subtweets and known non-subtweets because one does not already exist.
- Tweets are classified as either predicted subtweets or predicted non-subtweets.
- An automated program which accesses Twitter (a Twitter bot) interacts with subtweets in real time.

The Ground Truth Dataset

This project treats identification of subtweets as a text classification problem. Supervised machine learning on text through classification typically requires some **ground truth dataset** of how specific documents ought to be categorized prior to any actual machine learning.

- For this project, known subtweets and known non-subtweets were acquired using the Twitter search API with a query which exclusively selects tweets with **replies** which themselves *contain* or *exclude* the string **"subtweet"**.
- This generalization was developed with **call-out culture** in mind. A particular pattern was observed that Twitter users often call-out subtweets from their peers in order to ask if they are the target or complain about the very act of subtweeting.
- This method for data acquisition was a *fast and cheap* way to gather data for training the classifier, but a tweet is not necessarily a subtweet just because a user *happens* to reply to it with or without the string **"subtweet"**.

Whereas the ground truth dataset acquisition process necessarily relies on **replies**, the actual classification **deliberately excludes them** in favor of pure text classification in order to classify tweets live without the need for any users to have already interacted with them. **Figure 1** shows a tweet which is present in the ground truth dataset and a reply which was used to identify it as a subtweet.

Training & Testing the Classifier

- The **Naive Bayes** classification algorithm makes use of **Bayes Rule** to predict the likelihood that a particular set of features (e.g. words) belong to a particular class [1].
- It makes the **naive** assumption that the features are conditionally independent: the *presence* or *omission* of a particular feature does not change the likelihood of encountering other features within that class.

$$\mathrm{Pr(class|word)} = \frac{\mathrm{Pr(word|class)Pr(class)}}{P\mathrm{(word)}}$$

Naive Bayes computes the product of all the predicted probabilities for each word in the document. The greatest product computed across all the classes becomes the predicted class for that document.

- k-folds cross-validation was used to split apart **training** and **testing** sections of the ground truth dataset [2].
- For each fold k_i, the algorithm selects different sections of the entire dataset as the **training** and **testing** sets within that fold.
- The statistics on the performance of the classifier are computed for that fold alone and averaged across all folds to measure the overall performance of the classifier.

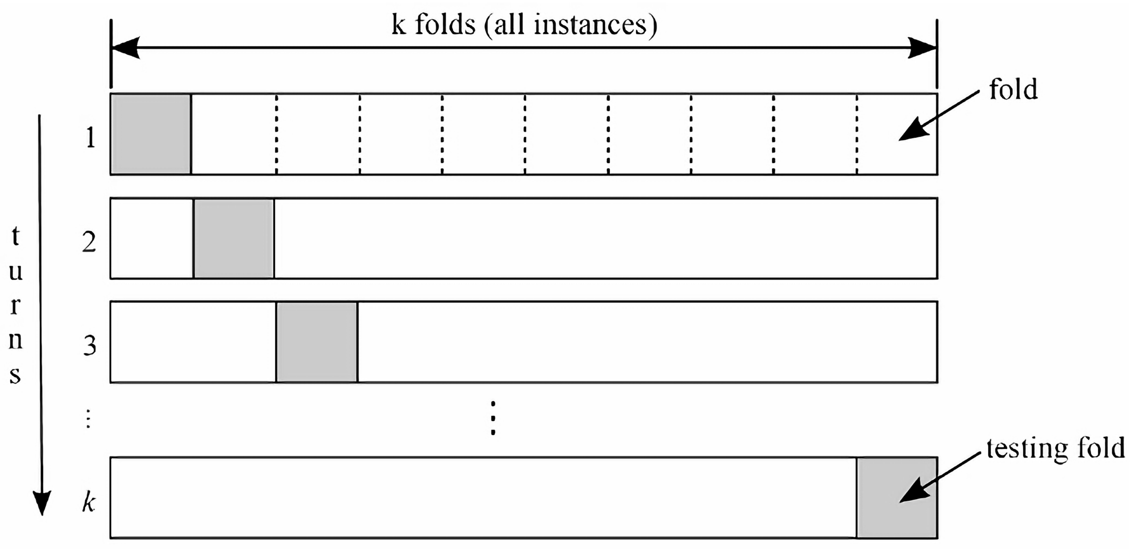

Figure 2: k-folds Cross-Validation Example [2]

To measure the performance of the classifier, it is necessary to keep track of how many **correct** and **incorrect** predictions it makes. These predictions are quantified in terms of **true positives**, **true negatives**, **false positives**, and **false negatives**.

- **True positives** (TP) are known subtweets which were **correctly** classified as predicted subtweets.
- **True negatives** (TN) are known non-subtweets which were **correctly** classified as predicted non-subtweets.
- **False positives** (FP) are known non-subtweets which were **incorrectly** classified as predicted subtweets.
- **False negatives** (FN) are known subtweets which were **incorrectly** classified as predicted non-subtweets.

Thus, the performance of the classifier is measured in terms of precision, recall, and F1 score.

$$\mathrm{Precision} = \frac{TP}{TP+FP} \qquad \mathrm{Recall} = \frac{TP}{TP+FN}$$

$$F_1 = \frac{2*(\mathrm{Precision}*\mathrm{Recall})}{\mathrm{Precision}+\mathrm{Recall}}$$

	Precision	Recall	F_1 Score
non-subtweets	0.7357	0.6988	0.7166
subtweets	0.7132	0.7490	0.7305

Table 1: Statistics Averaged Across 10 Folds of Cross-Validation

Confusion Matrix

Figure 3 is a confusion matrix which illustrates this performance in terms of raw counts and normalized over the entire ground truth dataset.

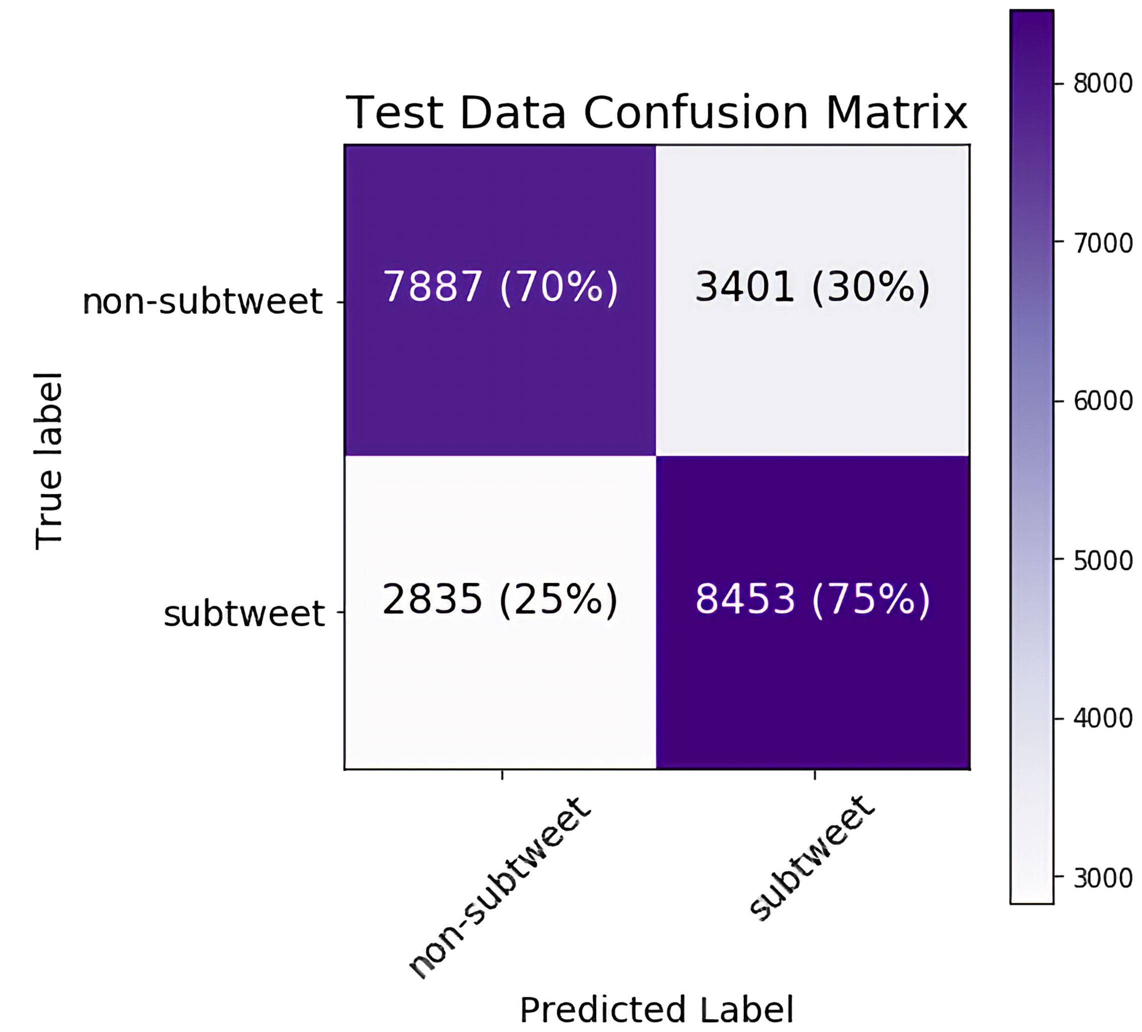

Figure 3: Accumulated Confusion Matrix for All 10 Folds

The Twitter Bot

After training and testing the classifier, it was utilized to create a Twitter bot which interacts with predicted subtweets in real time. It announces subtweets as they are posted in order to present covertly hurtful content as obviously hurtful in a public fashion. **Figure 4** shows a (censored) example of the bot quoting a user's tweet.

Figure 4: Example of the Twitter Bot Quoting a Tweet

References

[1] Harry Zhang, *The optimality of naive Bayes*, AA **1** (2004), no. 2, 3.

[2] Tomas Borovicka, Marcel Jirina Jr, Pavel Kordik, and Marcel Jirina, *Selecting representative data sets* (2012).