Who Tweeted That?

Yunqiang Pu 909662 Yizhou Wang 669026 Tzu-Tung Hsieh 818625

1. Introduction

Authorship attribution is the process to identify the author of the given texts [1]. Basically, it uses statistical methods to make the prediction in terms of the user pattern, emoticons or grammatical features. Its significance cannot be neglected as it has indicated a wide range of applications in various areas. In this report, distinctive approaches are discussed.

2. Datasets

There are generally two datasets available for the system development and test. *Train_tweets* provides the necessary collections for model training, which covers 328k tweets from 10k Twitter users. While for *test_tweets_unlabeled*, it gets over 35k tweets and their authorships are needed to be classified. It is obviously a tremendous task based on the size of the dataset and a lot of effort should also be put upon the system design to address the time or space complexity.

3. Preprocess and Feature Engineering

3.1 Manual Feature Extraction

To be prepared for the model development, preprocessing is crucial as it is fundamental for the further analysis [2]. Therefore, tweets are transformed and organized. For example, for each tweet, it is tokenized, lemmatized and lastly, presented in the lower case form. Certainly, relevant stop words, punctuation, and verb words are removed.

Original: Yoga is the cessation of mind. -Patanjali
Transform: yoga cessation mind -patanjali
All the processes are implemented and then saved into the pickle file in the local disk.

After the basic operations, different tweets' authors potentially possess different types of personality [3]. In this way, the unique patterns in tweets are needed to be considered.

Aspects	Pattern Form
Language	Sentiment; profanity; unique ratio; word count; sentence length; punctuation per tweet
Behaviour	Retweet; hashtag; emoticon; @mention
Other	URL: contain website or not; location: the potential city for the user

Table 1: Manual Potential Features

The table above illustrates the features that have been extracted in the feature engineering stage. The information selected could possibly be helpful candidate when making predictions for the authorship. The reason to select these features is quite intuitive as evidenced by the tweet length. To be more specific, someone is love to spend longer time recording personal life experience while others may not. Therefore, the length would be much longer.

3.2 Machine Feature Extraction

To manually extract the features needed for the model is not only difficult, but also inconvible or even not enough for the model building [4]. Therefore, in the later stage, Term frequency inverse document frequency (Tf-idf) and Bag of Words (BoW) are utilized for extracting more features to meet the expectation.

4. System Implementation

Various systems have been attempted and evaluated. The general methods implemented in the first place is to select features and then be applied into the computational models.

4.1 Feature Comparison

4.1.1 Manual Features

The features listed in the table before have been utilized and developed for the first model. Without any doubt, the linear model explains the data poorly, which is about 0.07%. For more information, the figure 1 compares linear model with different feature sizes.

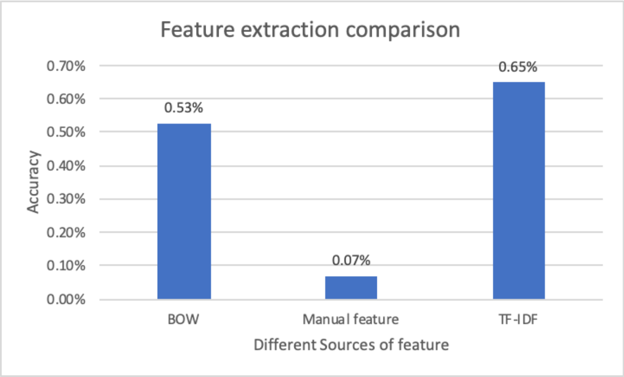

Figure 1: Feature Size Comparison

4.1.2 Machine Features

More features are selected by the Tf-idf and BoW models (50K) ranked by term frequency across the corpus [5].

It is obvious to notice that the feature size has a large impact on the performance when fitting in the linear model. More features make a lot of contributions to produce a much more accurate model regardless of BoW or Tf-idf.

4.2 BoW, Tf-idf and Feature Size

4.2.1 BoW and Tf-idf

BoW generally counts how many times a word appears in a document [6]. It doesn't consider the word sequence or semantic meanings. In other words, it seems to be the histogram that calculates the frequency for each word in the corpus. While for Tf-idf, it possesses similar characteristics but the apparent strength it brings is to identify the key word in the sentence [7].

The figure below clearly indicates that Tf-idf has a better performance no matter how many selected features. It can be said that roughly 3% performance difference is due to the capability of key word identification in Tf-idf model.

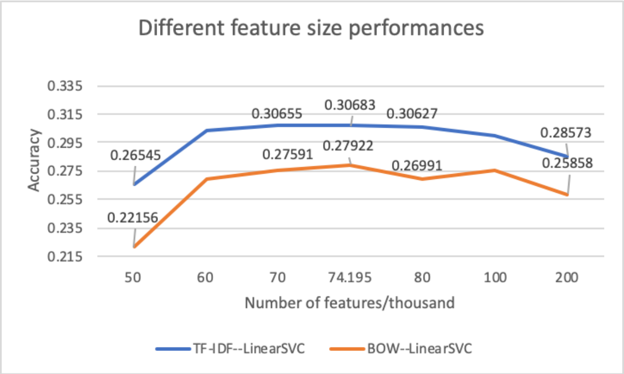

Figure 2: Different Feature Size Performances

4.2.2 Feature Size Impact

Still referring to Figure 2 above, it is interesting to find out that both of the models reach their peaks under the similar situation, which is to choose around 74K features. For the other number of features,it occasionally represents the signal of underfitting or overfitting problems, telling us the feature size is not enough or far enough than the optimal point shown in the graph. Therefore, the result discussed could give us the conclusion that it is beneficial to set the feature size ranging from 70K to 80K.

4.3 Processed vs Unprocessed

	BoW	Tf-idf
Processed	24.02%	27.87%
Unprocessed	26.102%	29.99%

Table 2: Processed vs Unprocessed

The table above demonstrates the unprocessed tweet outperform processed one approximately 2%. Jonathan J. Webster and Chunyu Kit have mentioned that processing sentences sometimes may not be required [8]. In this way, lower case, punctuation removal, lemmatization or even tokenization is no longer needed. Theoretically, all of the clues to categorize the authorship are embedded in the context. For instance, personal spelling errors could be the potential key to the classification problem. However, once the texts are processed, those valuable information then has been stripped off, which leads to worse results. Therefore, it is better to keep the tweets intact to produce more precise predictions.

4.4 Model Analysis

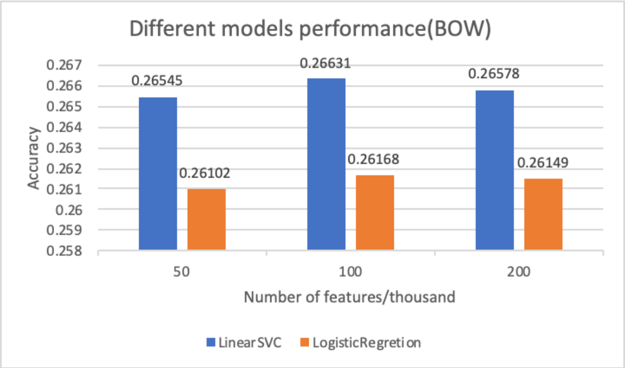

Figure 3: Model Performance

Considering the run time taken by the model, two models are explored and evaluated as the strong feasibility they have. Linear SVC tries to minimizes the squared hinge loss and owns the concept of one vs rest when dealing with the multi-class classification [9]. It has the strength like taking advantage of using liblinear as the estimator to reduce a large amount of time. In terms of logistic regression, instead of finding a particular optimal separating hyperplane, it is known as generalized linear regression, which aims to categorize depending on the probability [10]. The accuracy comparison indicates that linear SVC has pretty nice results if the feature extraction is defined.

The reason for that is probably high dimensional dataset. As the number of the author is nearly 10K, and the extracted features are approaching 70K or 80k. Handling with the data with this amount of the dimension is quite challenging and could be hard for the logistic regression. However, it is much easier for linear SVC [10].

5 Further Improvement

Doc2vec/Word2vec method can be very useful in predicting authorship. As they take account of the word relationship between each other [11]. For each author, it is good enough to develop a language model and put the word embeddings in the feature extraction process, then a more accurate model would be produced but it takes a lot more time and more computational resources than the model used in this report.

6 Kaggle Results and Improvement

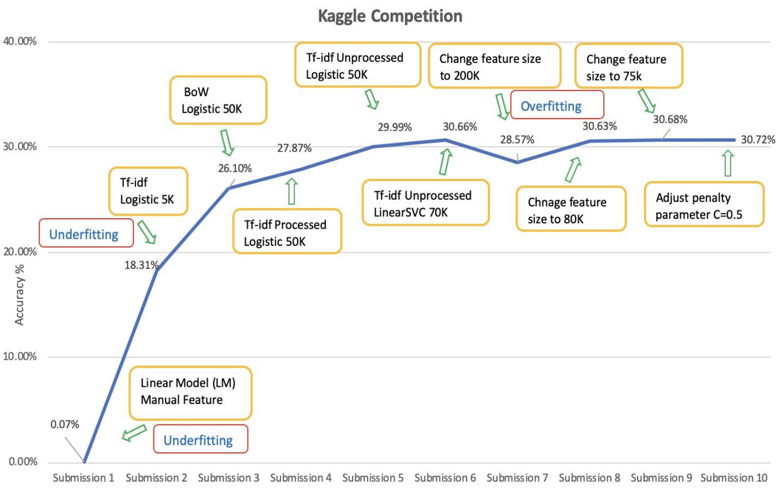

Figure 4: Kaggle Results

7 Conclusions

Based on the model implementation before, several conclusions could be drawn as follows:

1. Feature size performs better between 70k-80K, to possibly overcome the under/overfit problem.
2. Tf-idf is able to identify the sentence keyword, so results in a better grade, compared to BoW.
3. Linear SVC is capable of dealing with much higher dimensional data than logistic regression, especially suitable for this project's datasets.
4. Unprocessed tweets keep more informative data when doing the authorship classification instead of the processed one.
5. Changing penalty factor (C) to a smaller value would improve the data somehow as it looks for the large margin separating hyperplane, saying it allows more outliers to avoid overfit problem.
6. The feature extraction and model selection are extremely critical when building the model.

Therefore, Unprocessed Tf-idf with 75k features combined with linear SVC (C = 0.5) is chosen for our final model, which achieves 30.721%.

Reference

[1] Rosa, M., Luis, V. Pineda., Manuel, M. Gómez. & Paolo, R. Authorship Attribution Using Word Sequences (2006). *Springer, Berlin, Heidelberg.* 4225(1), 844-853. doi: 10.1007/11892755_87

[2] Daniel, T. Larose. & Chantal, D. Larose. Discovering Knowledge in Data, An Introduction to Data Mining (2014). *John Wiley & Sons Inc.* 2(1), 45-75.

[3] Michael, M. Tadesse., Hongfei, L., Bo, X. & Liang, Y. Personality Predictions Based on User Behavior on Facebook Social Media Platform (2018). *Dalian University of Technology*. 28(6), 1960-1965. doi:10.1109/ACCESS.2018.2876502

[4] Joachims, T. Text Categorization with Support Vector Machines: Learning with many Relevant Features (1998). *Springer, Berlin, Heidelberg.* 4(5), 137-142

[5] Scikit-learn Documentation. Retrieve from: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

[6] Zhang, Y., Rong, L. & Zhihua, Z. Understanding bag-of-words model: A statistical framework (2010). *International Journal of Machine Learning and Cybernetics.* 1(1), 43-52.

[7] Shahzad, Q. & Ramsha, A. Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents (2018). *International Journal of Computer Applications.* 181(1). 25-28.
doi: 10.5120/ijca2018917395

[8] Webster, J. J. & Kit, C. Tokenization as the Initial Phase in NLP. In COLING (1992). 4(3). *The 15th International Conference on Computational Linguistics.*

[9] Chia-Hua, H. & Chih-Jen, L. Large-scale Linear Support Vector Regression (2012). *Journal of Machine Learning Research.* 3324-3347.

[10] Diego, A. S., Jorge, I. V. & Juan C. Salazar. Comparison between SVM and Logistic Regression: Which One is Better to Discriminate (2012). *Revista Colombiana de Estadística*. 35(2). 223-237.

[11] Tomas, M., Kai, C., Greg, C. Jeffrey, D. Distributed Representations of Words and Phrases and their Compositionality. 1-3. Retrieve from https://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-compositionality.pdf