Text classification: TF-IDF vs word vectors

Jeremy Dale Coronia
College of Computer Studies
De La Salle University - Manila
Metro Manila, Philippines
jeremy_coronia@dlsu.edu.ph

Dr. Charibeth Cheng
College of Computer Studies
De La Salle University - Manila
Metro Manila, Philippines
charibeth.cheng@dlsu.edu.ph

***Abstract*—Text classifiers were trained on a binary class dataset and a multi-label dataset using Bag-of-Words and word vectors as feature extraction techniques. Moreover, in Bag of Words, comparisons were made between performing TF-IDF across the dataset versus TF-IDF per label. For word vectors, a comparison was also made to determine whether using a pre-trained word vector (GloVe) or training one from scratch would yield higher performance values. Text pre-processing techniques were also applied to clean the data before model building. Results indicate that the TF-IDF implementation across the whole dataset performed overall best in binary classification. Alternatively, pre-trained word vectors yielded the highest evaluated performance in the multi-label classification problem.**

***Index Terms*—natural language processing, text classification**

I. Introduction

This report details attempts on text classification from a binary class dataset, and a multi-label dataset. Four (4) classifiers were trained in total, two (2) classifiers for each dataset. A Bag-of-Words model and a model utilizing word vectors was implemented for each dataset. For both datasets, text pre-processing was done using natural language processing techniques before model building.

II. Description of the Dataset

For this experiment, 2 datasets were used, one (1) for each classification problem.

A. Binary Classification

The dataset used in binary text classification is found in Kaggle [1]. This dataset contains roughly 20 thousand Twitter user profile information, as well as other supplementary data such as an associated tweet, account profile and image. Twitter data is associated with a gender: female, male, and brand, which is considered as "non-human." Unknown genders are classified as "unknown." Figure 1 below shows the distribution below.

During data analysis, it was found that the gender distribution was more or less balanced. The "brand" value was omitted in this experiment, as was any unknowns in the column.

B. Multi-Label Classification

This dataset is also publicly available on Kaggle [2]. It contains 100 rows of environment-related news excerpts from a book. The dataset has a Ukranian and English version. Only the English version is used in this dataset.

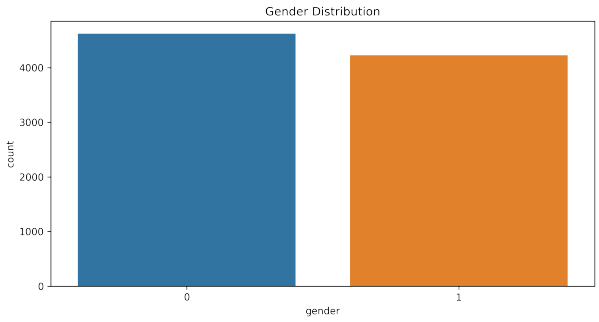

Fig. 1. The distribution of females (0) and males (1) in the dataset.

The dataset is comprised of the text excerpt itself, as well as five (5) binary features, which are as follows:

1) *env_problems*: Is the text about an environmental problem?
2) *pollution*: Is the text about environmental pollution?
3) *treatment*: Is the text about treatment plans or environmental technologies?
4) *climate*: Is the text about climate indicators?
5) *biomonitoring*: Is the text about biological, biotic monitoring in water or in a river basin?

All 5 [binary] feature columns contain only either 0 or 1 as its value per row. Figure 2 shows the distribution of said features across the dataset.

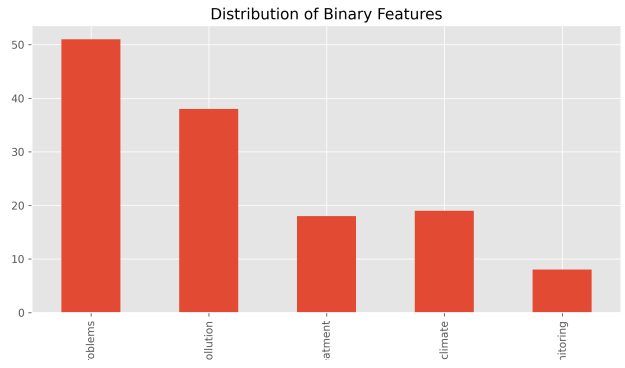

Fig. 2. The distribution of binary features in the dataset.

It can be seen in this figure that the dataset is skewed in

terms of features, and that the *env_problems* feature is represented the most while *biomonitoring* is the least represented.

III. Classifiers

Data was split into training set (60%) and testing set (40%). 5-fold cross validation was also performed to ensure that each class was well-represented in each fold. In particular, stratified cross validation was performed for the binary classifiers.

In evaluating the model's performance, F-score, recall, and precision are observed.

This experiment uses the following machine learning algorithms for all classifiers in the succeeding subsections:

1) Multinomial Naive Bayes (MNB).
2) Gaussian Naive Bayes (GNB).
3) Logistic Regression (LOG).

All classifiers were implemented using Scikit-learn, a machine learning library available in Python [3]. Text pre-processing, on the other hand, was implemented with the help of the Natural Language Toolkit [4].

The settings for the classifiers were kept to to Scikit-Learn's default configurations except for logistic regression, wherein the class weights were set to balanced and the solver was set to 'liblinear.'

A. Data Pre-processing

1) Binary Classification: For binary classification, the classifiers were trained in binary gender classification; namely, female and male. Although, excluding empty rows, there is a "brand" and "unknown" value in the *gender* column, this was ignored and dropped during data pre-processing.

The input for the model is the profile description or *description* column. The labels used are female and males, which are encoded into 0 and 1, respectively.

For the purposes of this experiment, the following conditions have been set to clean the dataset before model building.

1) All null rows found in description or gender will be dropped.
2) All rows containing the "unknown" or "brand" value in the *gender* column will be dropped.
3) All rows having a gender confidence less than 80% will be dropped.

The description column was then cleaned afterwards. The following steps were taken.

1) All text was converted to lowercase.
2) URLs and HTML tags were removed.
3) Emojis and punctuation were also removed.
4) Stop words are removed.
5) Text was tokenized, then lemmatized.

After data cleaning, there were 8847 rows left, roughly evenly distributed across both genders.

2) Multi-Label Classification: Much of the methods used for text pre-processing in binary classification is also used in multi-label classification.

The input for this model is are the text excerpts while the labels are the 5 binary features discussed above. There are only 100 rows in this dataset even after data pre-processing.

After text pre-processing, model building follows afterwards.

B. Bag of Words

For bag of words, the Term Frequency Inverse Document Frequency (TF-IDF) was used. This research also experimented only with 1-grams or unigrams. Max features were set to 4500.

During experimentation, a comparison was also made between implementing TF-IDF across the entire dataset for both labels (TF-1) versus applying TF-IDF for either labels first and then concatenating them (TF-2) prior to model building.

C. Word Vectors

For word vectors, GloVe [5] was used in working with a pre-trained word embedding. This experiment also compared the results between using a GloVe and training a word embedding from scratch.

D. Binary Relevance

For multi-label classification, because several labels may be found in any given input, binary relevance was used to adjust the machine learning models to the equivalent of several single-label classifiers per label. In other words, each single-label classifier only concerns itself with membership or non-membership to a given class. A possible drawback of this implementation is that it assumes that the 5 labels in the dataset do not have any dependencies.

IV. Results and Discussion

A total of 12 models were trained, three (3) per type of implementation. To determine the best models, F-score was used as basis.

A. Binary Classification

For the Bag-of-Words model, max features were set to 4500 because it resulted in the highest observed performance. Moreover, applying TF-IDF per class label (TF-2) resulted in worse performances than when applying it across the entire dataset (TF-1). In this type of classifier, it can be seen that the Multinomial Naive Bayes model performed best in terms of F-score. Table 1 below shows the comparison of results per methodology.

Algorithm	TF-1			TF-2		
	F1	PRE	REC	F1	PRE	REC
MNB	**66.10%**	67.03%	66.60%	49.58%	49.90%	50.38%
GNB	61.24%	62.38%	62.03%	48.75%	50.19%	49.53%
LOG	65.85%	66.13%	66.06%	**50.54%**	50.55%	50.75%

TABLE I
TABLE OF VALUES FOR TF-1 AND TF-2 IN BINARY CLASSIFICATION.

Out of all trained classifiers in TF-1, 66.10% was the highest recorded performance; however, performance was observed to have dipped when feature extraction was performed for per label. The highest evaluated performance in TF-2 is 50.54%.

Algorithm	From Scratch			GloVe		
	F1	PRE	REC	F1	PRE	REC
MNB	**52.05%**	54.31%	54.28%	**50.86%**	53.94%	53.86%
GNB	35.56%	54.38%	51.82%	36.87%	51.32%	51.79%
LOG	46.86%	53.07%	52.90%	52.92%	55.35%	55.13%

TABLE II
TABLE OF COMPARISON OF VALUES OF TRAINING WORD VECTORS IN BINARY CLASSIFICATION.

Alternatively for word vectors, results using a pre-trained word embedding (i.e., GloVe) were compared against training one from scratch. Table 2 shows the comparison of results.

It can be seen here that training a word embedding from scratch seem to result in a better performance with a 52.05% recorded F-score compared against using GloVe, which has a score of 50.86%. This seems to indicate with training one from scratch, the classifier learns more from the data it is fed with. The performance loss may also be attributed to the pre-trained word vectors used, although this hypothesis was not explored further with other available word vectors online.

B. Multi-Label Classification

For the Bag-of-Words model, the max features were set to 50. It can be seen that logistic regression classifier performed the best in multi-label classification out of all other models, in terms of F-score. Table 3 below shows the comparison of values.

Algorithm	TF-1			TF-2		
	F1	PRE	REC	F1	PRE	REC
MNB	52.83%	56.67%	52.50%	8.750%	12.50%	7.083%
GNB	54.17%	51.04%	60.00%	**31.18%**	34.96%	34.58%
LOG	**54.33%**	48.75%	63.75%	25.42%	25.63%	30.00%

TABLE III
TABLE OF VALUES FOR TF-1 AND TF-2 IN MULTI-LABEL CLASSIFICATION.

It can be seen in this table that in TF-1, the highest recorded performance is 54.33%, while in TF-2, the overall performance dipped to 31.18%, again indicating at least that performing feature extraction per label would not benefit in increased performance.

Word Vectors. For multi-label classification, a comparison was also observed from using a pre-trained word embedding against making one from scratch. (this table of values is not updated yet).

Algorithm	From Scratch			GloVe		
	F1	PRE	REC	F1	PRE	REC
MNB	**35.58%**	69.24%	35.71%	39.17%	36.25%	45.42%
GNB	33.93%	57.08%	25.33%	**60.92%**	50.60%	91.07%
LOG	15.33%	18.75%	13.33%	38.75%	35.42%	49.17%

TABLE IV
TABLE OF COMPARISON OF VALUES OF TRAINING WORD VECTORS IN MULTI-LABEL CLASSIFICATION.

Table 4 above shows the table of values.

V. CONCLUSION

This report detailed experiments with binary and multi-label classification problems on small datasets. In binary classification, it can be seen that the highest recorded performance across both implementations is in Bag-of-Words, indicating that counting the term frequency of words and its relevance to the dataset is better in this instance than representing words as vectors. On the other hand, using pre-trained word vectors to represent input in multi-label classification seemed to result in the highest recorded performance.

It should be noted, however, that the dataset for the multi-label performance is extremely small, having only 100 rows. The same may also apply for the binary classification corpus, which had only roughly 8000 rows. This means that overfitting may occur, although that only simple classifier models were used may have helped in mitigating the chances of overfitting.

Another thing that must also be noted is that even after text pre-processing, some text may still be indecipherable during training and model building (e.g., "ab" was observed in the processed text), which may affect model performance. Other outlying factors that may have affected model performance are errors in encoding, textspeak and intentionally misspelled words, which is prevalent in the first dataset, as well as texts in other languages. Given this, further text cleaning and pre-processing may be needed to improve performance of the classifiers.

REFERENCES

[1] King, E. (2016). *Twitter user gender classification, Version 1*. https://www.kaggle.com/crowdflower/twitter-user-gender-classification

[2] Mokin, V. (2020). *NLP: Reports and news classification, Version 1*. https://www.kaggle.com/vbmokin/nlp-reports-news-classification?select=water%5C_problem_nlp_ua_for_Kaggle_100.csv

[3] Scikit-learn: Machine learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[4] Bird, S., Loper, E., & Klein, E. (2009). *Natural language processing with python.* O'Reilly Media Inc.

[5] Pennington, J., Socher, R., & Manning, C. D. (2015). *GloVe: Global vectors for word representation*. https://nlp.stanford.edu/projects/glove/