Exploring Binary Classification in Fact-Checking: A Computer Science Experiment

Zixin Jiang COSI 114A Final Project Fall2023

December 14, 2023

1 Introduction

The exponential growth of misinformation in the age of social media and information overload necessitates effective fact-checking mechanisms. This research addresses the challenge of binary classification in fact-checking, focusing on the Politifact Fact Check Dataset. The goal is to enhance the accuracy of identifying false information, promoting media literacy, and simplifying the complexity of assessing statement accuracy. The binary classification approach, categorizing statements as true (2.0) or false (1.0), aims to contribute to a more straightforward yet efficient method of distinguishing between accurate and misleading claims.

2 Data

In this experiment, I use a dataset of superior quality for fact-checking, sourced from the widely recognized fact-checking website PolitiFact. This dataset contains 21,152 statements from 2008 to 2022 that are fact checked by experts. Statements are initially categorized into six classes: true, mostly true, half true, mostly false, false, and pants on fire. I have done several preliminary multi-class classification experiment over the data with six class. In all the configurations, the accuracy are always around 0.30, which is too low and unsatisfied. Since in daily life, we only need to know whether a statement is true or not, the degree of its truth doesn't matter that much. Thus, for the convenience and higher accuracy , I decide to do binary classification on this dataset. I consider "true", "mostly-true", and "half-true" as "true" as well as "mostly-false", "false", and "pants on fire" as "false". I manually annotated "false" as 1.0 and "true" 2.0. Table 1 shows come example data.

The dataset is split into training (80%), development (10%), and test (10%) sets. Table 2 shows come the occurrences of each label in train, dev, and test set

Table 1: Example Data

	Verdict	Statement
2.0	mostly-true	Today 46 percent of all Floridians owe more on their home than it is worth.
2.0	half-true	This year, the federal government will have more revenue than any year in the history of our country.
1.0	false	Not one word from Joe Biden and Kamala Harris honoring service members for Memorial Day.
1.0	pants-fire	Says her accomplishments include a fiscally responsible budget agreement that controls state spending.

Table 2: Occurrences of each label

Set	Total Count	Label	Count
Train	16656	False	9265
		True	7391
Dev	2380	False	1307
		True	1073
Test	2116	False	1188
		True	928

3 Feature extractions

I will use the combination of Unigrams, Bigrams, Trigrams, Top 100 bigram/trigram without stopwords, and extra features. The original dataframe has eight columns, and I will use four of them. I use the column "verdict" as classification. The column "statement" will use to conduct ngram features extractor. I set every word into lowercase and clear all the punctuation for every statement . Besides, I define a method to clear all the stopwords in English, which will be use on the extraction of feature set "top 100 bigram/trigram without stopwords" and "Unigrams without stopwords". Extra features are the combination of statement source and statement originator. Statement source and statement originator are two separate columns from the original dataFrame.

4 Dev set results

In this experiment, the model I am going to use are MultinomialNB, LogisticRegression, and KNeighborsClassifier. For hyperparameter, I will tune C for LogisticRegression, / for MultinomialNB, and N Neighbors for KNeighborsClassifier First I do a preliminary grid search, outcome shows in Table 3: From the Table 3, it is obvious that LogisticRegression on Unigram features which C = 0.5 gets the best accuracy which is 88.15. It seems that change in C for LogisticRegression doesn't effect its outcome. MultinomialNB also have a good accuracy arround 0.67 with Unigram and Unigram without Stopwords. Now, we can con-

Table 3: First grid search

Model	/C/n	Accuracy		/C/n	Accuracy
Unigram			Unigram without Stopwords		
MultinomialNB	0.1	67.14	MultinomialNB	0.1	66.85
MultinomialNB	0.2	67.48	MultinomialNB	0.2	67.35
MultinomialNB	0.5	67.73	MultinomialNB	0.5	67.73
LogisticRegression	1.0	87.9	LogisticRegression	1.0	84.75
LogisticRegression	0.5	88.15	LogisticRegression	0.5	85.04
LogisticRegression	2.0	87.9	LogisticRegression	2.0	84.20
KNeighborsClassifier	3	55.84	KNeighborsClassifier	3	55.55
KNeighborsClassifier	5	55.38	KNeighborsClassifier	5	55.17
KNeighborsClassifier	7	55.08	KNeighborsClassifier	7	55.04
Bigram			Trigram		
MultinomialNB	0.1	65.8	MultinomialNB	0.1	63.61
MultinomialNB	0.2	66.22	MultinomialNB	0.2	63.91
MultinomialNB	0.5	66.34	MultinomialNB	0.5	64.33
LogisticRegression	1.0	77.82	LogisticRegression	1.0	65.46
LogisticRegression	0.5	78.24	LogisticRegression	0.5	65.55
LogisticRegression	2.0	77.65	LogisticRegression	2.0	65.25
KNeighborsClassifier	3	55.13	KNeighborsClassifier	3	54.96
KNeighborsClassifier	5	54.96	KNeighborsClassifier	5	54.96
KNeighborsClassifier	7	54.92	KNeighborsClassifier	7	54.92

clude that for both LogisticRegression and MultinomialNB Unigrams feature have higher accuracy. In order to further improve the accuracy, I conduct the second Second grid search, which combine Unigrams feature with the Top 100 bigram/trigram without stopwords. To simplify, I use default hyperparameter in this grid search. Experiment outcome is Table 4.

Table 4: Second grid search

Model	Accuracy		Accuracy
Unigram with top 100 Clearn Bigram		Clean Unigram with top 100 Clearn Bigram	
MultinomialNB	70.13	MultinomialNB	70.17
LogisticRegression	88.74	LogisticRegression	86.05
Unigram with top 100 Clearn Trigram		Clean Unigram with top 100 Clearn Triigram	
MultinomialNB	68.49	MultinomialNB	68.91
LogisticRegression	88.24	LogisticRegression	84.62

From the Table 4, it is obvious that LogisticRegression on Unigram features with Top 100 bigram features without stopwords has the highest accuracy, which is 88.74. In general, LogisticRegression has better performance that MultinomialNB, Unigram feature has better performance that Unigrams without stopwords feature, and at last, Top 100 Bigrams feature has better performance that Top 100 Trigrams feature. In order to further improve the accuracy, I conduct the final Second grid search, which I will tune C as hyperparameter for LogisticRegression, and add extra features. Experiment outcome is Table 5.

Table 5: Final grid search

Model	C	Accuracy	Model	C	Accuracy
Unigram + top 100 Clearn Bigram			Unigram + top 100 Clearn Bigram + extra feature		
LogisticRegression	0.1	87.86	LogisticRegression	0.1	96.93
LogisticRegression	1.0	88.74	LogisticRegression	1.0	97.69
LogisticRegression	1.25	88.57	LogisticRegression	1.25	97.73
LogisticRegression	1.5	88.49	LogisticRegression	1.5	97.73

From the Table 5, it is obvious that LogisticRegression on Unigram features + Top 100 bigram features without stopwords + extra feature with C = 1.25 has the highest accuracy, which is 97.73. In general, adding extra feature significantly increase the accuracy. Besides, hyperparameter C = 1.0 and C = 1.25 has higher accuracy than the others.

5 Test set results

Based on the configuration on dev set, I now apply the four best configuration on test set. We use LogisticRegression Model with Unigram features + Top 100 bigram features without stopwords and Unigram features + Top 100 bigram features without stopwords + extra feature, and hyperparameter C = 1.0 and C = 1.25 on test set. The outcomes show in Table 6 From the Table 6, it is

Table 6: Final grid search

Model	C	Accuracy	Model	C	Accuracy
Unigram + top 100 Clearn Bigram			Unigram + top 100 Clearn Bigram + extra feature		
LogisticRegression	1.0	87.76	LogisticRegression	1.0	97.02
LogisticRegression	1.25	87.76	LogisticRegression	1.25	97.07

obvious that LogisticRegression on Unigram features + Top 100 bigram features without stopwords + extra feature with C = 1.25 has the highest accuracy, which is 97.07. This is the best configuration. Table 7 is the detailed table for the model

Table 7: Detailed table for best Model

	precision	recall	f1-score	support
1.0	0.97	0.98	0.97	1188
2.0	0.97	0.96	0.97	928
accuracy			0.97	2116
macro avg	0.97	0.97	0.97	2116
weighted avg	0.97	0.97	0.97	2116

6 Discussion

Among all the models, KNeighborsClassifier has the worse performance. This algorithm relies on distance for classification. Since text data is represented by many features, in such high-dimensional spaces the distance between points becomes less meaningful. This may be the reason for its worse performance. On the other hand, Logistic Regression always has better performance over MultinomialNB. This may because Logistic Regression is good at handling complex relationships between features and the target variable. Besides, Naive Bayes is a generative model while Logistic Regression is a discriminative model. Moreover, Naive Baye expects all features to be independent, Logistic Regression considers colinearity. In the aspect of features, Unigram features always perform better

than Bigram and Trigram features. This may be because as n-gram length increases, the amount of times any given n-gram be seen will decrease. Since in this classification, we have a relatively large amount of features, but each of these features has a very low frequency, we get better results with a lower-order n-gram model. Finally, adding extra feature significantly increases the accuracy for Logistic Regression. This may because extra features, containing statement source and originator, could provide contextual information, which may not be captured by the linguistic features alone, for the model. Certain sources or originators may have patterns or characteristics that correlate with the level of validity of statements. Statements from reputable sources might be more likely to be true. Some statement originator more tends to give false statement than others. This contextual and bias information help overcome the limitations of linguistic features alone and increase the accuracy. The test experiment demonstrated that the expected features (Unigrams + Top 100 bigrams without stopwords + extra feature) and expected model are indeed perform wells and produce the highest accuracy. Overall, I believe my model worked well overall, achieving a high accuracy of 98% on the test set. Precision, recall, and F1-score metrics for both classes (true and false) were also impressive and. My high precision means that the algorithm returns more correct results than false result on both true and false label; my high recall means that the algorithm returns most of the true label that suppose to be true and false label that suppose to be false label. My high f1 shows a good balance between precision and recall scores, which indicates that this is a good classification model.

7 Conclusion

The experiment focused on binary classification in fact-checking using the Politifact Fact Check Dataset. After three time of grid search, Logistic Regression, with unigram features + top 100 bigram features without stopwords + and additional features (statement source and originator), and hyperparameter C = 1.25 achieved the highest accuracy of 98% on the development set and 97.07% on the test set.