CS579

Online Social Network Analysis

Fake News Classification

Project Report

Group 60

Swathi Puskoori A20513538

Sri Sai Teja Narra A20514763

Under Professor. Kai Shu

Contents | Page

1. Introduction	3
2. Problem Statement	3
3. Proposed Solution	3
4. Data Preprocessing	4
5. Feature Extraction	5
6. Classification Models	5
7. Validation Metrics	6
8. Results	6
9. Conclusion	9
10. References	10

Figures

1. Fig 4.1 Data before preprocessing
2. Fig 4.2 Data after Preprocessing
3. Fig 8.1 Results using Naïve bayes classifier.
4. Fig 8.2 Results using Decision Trees
5. Fig 8.3 Results using Logistic Regression

1. Introduction

Social media has become one of the major resources for people to obtain news and information. For example, it is found that social media now outperforms television as the major news source. However, because it is cheap to provide news online and much faster and easier to disseminate through social media, large volumes of fake news or misinformation are produced online for a variety of purposes, such as financial and political gain. The extensive spread of fake news/misinformation can have a serious negative impact on individuals and society: (i) breaking the authenticity balance of the news ecosystem; (ii) intentionally persuading consumers to accept biased or false beliefs; and (iii) changing the way people interpret and respond to real news and information. Therefore, it is important to detect fake news and misinformation in social media.

2. Problem Statement

We formally define the task as follow. Given the title of a fake news article A and the title of a coming news article B, participants are asked to classify B into one of the three categories:

• agreed: B talks about the same fake news as A.
• disagreed: B refutes the fake news in A.
• unrelated: B is unrelated to A.

3. Proposed Solution

The proposed solution is a classification model that predicts whether two news headlines are referring to the same news story or not. The solution involves:

- Data Preprocessing: The raw news headlines are preprocessed by converting them to lowercase, tokenizing them into words, removing stop words, and stemming the remaining words.

- Feature Extraction: The preprocessed news headlines are transformed into numerical features using the Tfidf Vectorizer from scikit-learn. This extracts the important words from the text and assigns weights to them based on their frequency and importance.
- Model Training: The logistic regression algorithm is used to train the classification model on the extracted features and corresponding labels (i.e., whether two headlines are the same or not).
- Model Evaluation: The trained model is evaluated on a validation set to measure its accuracy.
- Prediction: Finally, the trained model is used to predict whether the test set headlines refer to the same news story or not.

4. Data Preprocessing

We use NLTK (Natural Language Toolkit) to preprocess the data, the following are the preprocessing steps:

- Convert text to lowercase: This step is done to ensure consistency in the text. If some words are in uppercase and others are in lowercase, it can create confusion for the model. Lowercasing all the words ensures that the model treats the words equally.

- Tokenize text into words: Tokenization refers to splitting the text into individual words. This step is important because most natural language processing (NLP) algorithms work with individual words, not full sentences or paragraphs.

- Remove stop words: Stop words are common words that don't add much meaning to the text, such as "the," "and," "of," etc. Removing stop words helps to reduce the size of the data and make the processing faster.

- Stem the words: Stemming is the process of reducing words to their root form. For example, the words "running" and "run" have the same root word "run". Stemming helps to reduce the number of unique words in the data, which can make the processing faster and reduce the size of the model.

- Fill any missing values: If there are any missing values in the data, they need to be filled in before processing. This is done by replacing the missing values with an empty string.

- Save preprocessed data to CSV files: After preprocessing the data, save the data into a csv files.

	A	B	C	D	E	F
1	id	tid1	tid2	title1_en	title2_en	label
2	195611	0	1	There are two new old-age insurance benefits for old people in rural areas. Have you got them?	Police disprove "bird's nest congress each person gets 50,000 yuan" still old people insist on going to beijing	unrelated
3	191474	2	3	"If you do not come to Shenzhen, sooner or later your son will also come." In less than 10 years, Shenzhen per capita GDP will exceed Hong Kong.	Shenzhen's GDP outstrips Hong Kong? Shenzhen Statistics Bureau dismisses rumors: only the gap is narrowing	unrelated
4	25300	2	4	"If you do not come to Shenzhen, sooner or later your son will also come." In less than 10 years, Shenzhen per capita GDP will exceed Hong Kong.	The GDP overtopped Hong Kong? Shenzhen clarified: a little bit more......	unrelated
5	123757	2	8	"If you do not come to Shenzhen, sooner or later your son will also come." In less than 10 years, Shenzhen per capita GDP will exceed Hong Kong.	Shenzhen's GDP overtakes Hong Kong? Bureau of Statistics refutes rumor: Unsurpass but the gap shrinks again	unrelated
6	141761	2	11	"If you do not come to Shenzhen, sooner or later your son will also come." In less than 10 years, Shenzhen per capita GDP will exceed Hong Kong.	Shenzhen's GDP outpaces Hong Kong? Defending Rumors: The gap has narrowed yet again	unrelated
7	132794	6	15	"How to discriminate oil from gutter oil by means of garlic.	It's very practical to use a single piece of garlic to distinguish oil from oil! "	agreed
8	126388	6	17	"How to discriminate oil from gutter oil by means of garlic.	Differential gutter oil can be identified with a single piece of garlic.	agreed
9	6314	6	18	"How to discriminate oil from gutter oil by means of garlic.	stir-fried garlic to identify gutter oil	agreed
10	162344	7	14	It took 30 years of cooking oil to know that one piece of garlic is easy to spot.	A single piece of garlic can spot gutter oil? Come on! Do the following to keep you out of the gutter oil.	unrelated
11	28221	7	16	It took 30 years of cooking oil to know that one piece of garlic is easy to spot.	Use a garlic to distinguish oil from oil, very practical!	agreed
12	255346	7	18	It took 30 years of cooking oil to know that one piece of garlic is easy to spot.	stir-fried garlic to identify gutter oil	agreed
13	12412	7	39141	It took 30 years of cooking oil to know that one piece of garlic is easy to spot.	Quick identification of gutter oil, please put your oil in the refrigerator for 2 hours	unrelated
14	160593	9	10	"if you eat durian, you will kill yourself if you eat it wrongly!"	Durian can't eat with anything, it's the same as coffee, it's heart disease. "	unrelated
15	36755	10	88182	Durian can't eat with anything, it's the same as coffee, it's heart disease. "	The durian will die with milk? The durian can't eat with anything.	unrelated
16	174888	12	13	"Frog frog? It's a fertility test! Let's play" Jewel V "."	A store in xianning contains "cotton"? A multi-agency association in chongyang university	unrelated

4.1 Data before preprocessing

	A	B	C	D	E	F
1	id	tid1	tid2	title1_en	title2_en	label
2	195611	0	1	two new old-ag insur benefit old peopl rural area . got ?	polic disprov `` bird 's nest congress person get 50,000 yuan '' still old peopl insist go beij	unrelated
3	191474	2	3	`` come shenzhen , sooner later son also come . '' less 10 year , shenzhen per capita gdp exceed hong kong .	shenzhen 's gdp outstrip hong kong ? shenzhen statist bureau dismiss rumor : gap narrow	unrelated
4	25300	2	4	`` come shenzhen , sooner later son also come . '' less 10 year , shenzhen per capita gdp exceed hong kong .	gdp overtop hong kong ? shenzhen clarifi : littl bit	unrelated
5	123757	2	8	`` come shenzhen , sooner later son also come . '' less 10 year , shenzhen per capita gdp exceed hong kong .	shenzhen 's gdp overtak hong kong ? bureau statist refut rumor : unsurpass gap shrink	unrelated
6	141761	2	11	`` come shenzhen , sooner later son also come . '' less 10 year , shenzhen per capita gdp exceed hong kong .	shenzhen 's gdp outpac hong kong ? defend rumor : gap narrow yet	unrelated
7	132794	6	15	`` discrimin oil gutter oil mean garlic .	's practic use singl piec garlic distinguish oil oil ! ``	agreed
8	126388	6	17	`` discrimin oil gutter oil mean garlic .	differenti gutter oil identifi singl piec garlic .	agreed
9	6314	6	18	`` discrimin oil gutter oil mean garlic .	stir-fri garlic identifi gutter oil	agreed
10	162344	7	14	took 30 year cook oil know one piec garlic easi spot .	singl piec garlic spot gutter oil ? come ! follow keep gutter oil .	unrelated
11	28221	7	16	took 30 year cook oil know one piec garlic easi spot .	use garlic distinguish oil oil , practic !	agreed
12	255346	7	18	took 30 year cook oil know one piec garlic easi spot .	stir-fri garlic identifi gutter oil	agreed
13	12412	7	39141	took 30 year cook oil know one piec garlic easi spot .	quick identif gutter oil , pleas put oil refriger 2 hour	unrelated

4.2 Data after preprocessing

5. Feature Extraction

- scikit-learn (sklearn): Scikit-learn is a popular machine learning library for Python that provides tools for data mining, data analysis, and machine learning algorithms. It includes a wide range of machine learning models, preprocessing functions, feature selection methods, and evaluation metrics. Sklearn is built on top of NumPy, SciPy, and matplotlib.

- TF-IDF (Term Frequency-Inverse Document Frequency): TF-IDF is a widely used feature extraction method in information retrieval and text mining. It measures the importance of each word in a document by calculating its frequency in the document and inversely scaling it by its frequency in the corpus (all documents). The term frequency (TF) component measures how frequently a word appears in a document, while the inverse document frequency (IDF) component measures how important the word is in the corpus. The resulting TF-IDF matrix represents each document (in this case, a pair of titles) as a vector of numerical features that can be used for machine learning.

6. Classification Model Used

Logistic regression is a widely used method for binary classification tasks, where there are only two possible labels. In this case, the model is trained on the preprocessed and feature-extracted training data. The features extracted from the text are used to make predictions about the label. The logistic regression model learns to predict the label based on the features extracted from the text.

During the training process, 20% of the training data is reserved as the validation set. The validation set is used to evaluate the model's performance during training. The model is evaluated on the validation set to determine its accuracy, which is the percentage of correctly predicted labels.

Once the model is trained, it can be used to predict the labels for new data. The accuracy of the model can be further evaluated using other evaluation metrics such as precision, recall, F1-score, etc.

Other classification models used:

- Multinomial Naïve bayes
- Decision Trees

7. Validation Metrics

- Accuracy: This is a measure of how well a classifier correctly identifies true positives and true negatives out of all the samples it is predicting on. It is defined as the ratio of correct predictions to the total number of predictions made.

- Precision: Precision measures the proportion of true positives in the total number of predicted positives. It is defined as the ratio of true positives to the sum of true positives and false positives.

- Recall: Recall measures the proportion of true positives in the total number of actual positives. It is defined as the ratio of true positives to the sum of true positives and false negatives.

- F1-score: F1-score is a measure of the balance between precision and recall. It is the harmonic mean of precision and recall and is defined as 2*(precision * recall) / (precision + recall).

- Support: Support is the number of occurrences of each class in the dataset.

8. Results

- **Multinomial Naïve Bayes:**

```
Validation Accuracy for Multinomial Naive Bayes: 0.7438242118192985
Confusion matrix:
 [[ 5310     5  9498]
 [   10    63  1248]
 [ 2368    10 32777]]
              precision    recall  f1-score   support

      agreed       0.69      0.36      0.47     14813
   disagreed       0.81      0.05      0.09      1321
   unrelated       0.75      0.93      0.83     35155

    accuracy                           0.74     51289
   macro avg       0.75      0.45      0.47     51289
weighted avg       0.74      0.74      0.71     51289



Process finished with exit code 0
```

8.1 Results for Multinomial Naïve Bayes Classifier

- **Decision Trees**

```
Validation accuracy for Decision Trees: 0.7811378880049914
Validation performance metrics:
              precision    recall  f1-score   support

     agreed       0.67      0.67      0.67     14813
  disagreed       0.40      0.36      0.38      1321
  unrelated       0.84      0.84      0.84     35154

   accuracy                           0.78     51288
  macro avg       0.64      0.63      0.63     51288
weighted avg      0.78      0.78      0.78     51288

[[ 9992    51  4770]
 [   62   480   779]
 [ 4885   678 29591]]
```

8.2 Results using Decision Trees

- **Logistic Regression**

```
Validation accuracy for Logistic Regression : 79.61%
              precision    recall  f1-score   support

     agreed       0.72      0.60      0.66     14813
  disagreed       0.75      0.18      0.29      1321
  unrelated       0.82      0.90      0.86     35155

   accuracy                           0.80     51289
  macro avg       0.76      0.56      0.60     51289
weighted avg      0.79      0.80      0.79     51289

[[ 8918    11  5884]
 [   45   235  1041]
 [ 3409    69 31677]]
```

8.3 Results using Logistic Regression

- Submission.csv

	A	B
1	id	label
2	256442	unrelated
3	256443	unrelated
4	256444	unrelated
5	256445	unrelated
6	256446	unrelated
7	256447	unrelated
8	256448	unrelated
9	256449	unrelated
10	256450	unrelated
11	256451	unrelated
12	256452	agreed
13	256453	agreed

8.4 Submission.csv output.

- Comparing all the models

Classifier		Precision	Recall	F1-Score	Support	Accuracy
Naïve Bayes	Agreed	0.69	0.36	0.47	14813	74%
	Disagreed	0.81	0.05	0.09	1321	
	Unrelated	0.75	0.93	0.83	35155	
Decision Trees	Agreed	0.67	0.67	0.67	14813	78%
	Disagreed	0.40	0.36	0.38	1321	
	Unrelated	0.84	0.84	0.84	35154	
Logistic Regression	Agreed	0.72	0.60	0.66	14813	**80%**
	Disagreed	0.75	0.18	0.29	1321	
	Unrelated	0.82	0.90	0.86	35155	

9. Conclusion

We have employed various machine learning algorithms to develop a model that can accurately predict the classification of news article B as either real or fake using the given news article A. After evaluating the performance of different algorithms, we found that logistic regression achieved the highest accuracy compared to other models. Therefore, we have selected logistic regression as the most suitable model for this task.

There is significant potential for further exploration using neural networks and deep learning techniques in this project. Some possible avenues for future research include:

- Exploring additional techniques for feature engineering to enhance the accuracy and performance of the model.
- Adapting the model to handle new types of fake news that may arise in the future.
- Developing real-time detection systems that can quickly identify and classify news articles as real or fake as soon as they are published.
- Investigating ensemble methods to further improve the prediction accuracy and reduce overfitting.

By pursuing these potential future scopes, we can improve the effectiveness and practicality of the model and contribute to the ongoing efforts to combat the spread of misinformation in the news.

10. References

- https://pypi.org/project/stop-words/
- https://www.nltk.org/_modules/nltk/stem/wordnet.html
- https://scikit-learn.org /stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html
- https://scikitlearn.org/stable/modules/generated/sklearn.metrics.pairwise.cosine_similarity.ht ml
- https://iq.opengenus.org/document-similarity-tf-idf/
- https://towardsdatascience.com/natural-language-processing-feature-engineering-using-tf-idf-e8b9d00e7e76
- https://iq.opengenus.org/document-similarity-tf-idf/