Natural Language Processing : Project 1

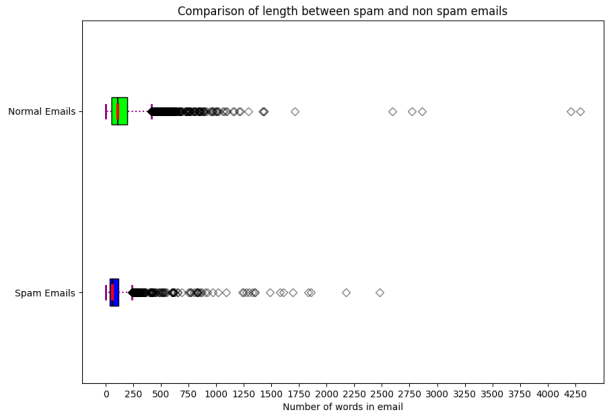
ANNE-LAURE TETTONI and JULIO CESAR PEREZ DURAN

ACM Reference Format:
Anne-Laure Tettoni and Julio Cesar Perez Duran. 2022. Natural Language Processing : Project 1. 1, 1 (July 2022), 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INSIGHT FROM DATA

We extracted some insights from the given data. Here is what we found :
Number of spam emails in data :**1368**
Number of non spam emails in data : **4360**
Number of unique words in spam emails: **8693**
Number of unique words in non spam emails: **8376**
Number of unique words across all emails: **14081**
The most common words in spam and non spam email respectively can be seen in table 1 and table 2.

We also used a box-plot to see how the lengths between spam and non-spam email are spread out, and this is what we found :



The average number of words in a spam email is : **124.6**
The average number of words in a non-spam email is : **157.9**

One can see that spam emails tend to be shorter. The lengths of normal emails are more spread out than the lengths of spam emails, indicating that they are more varied and less predictable.

Authors' address: Anne-Laure Tettoni; Julio Cesar Perez Duran.

XXXX-XXXX/2022/7-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Table 1. Most common words in spam email

Words	Frequency
one	1613
zero	1449
company	1065
com	998
business	891
email	858
five	788
two	771
three	753
information	748
money	662
mail	651
get	612
free	606
http	603
time	581
please	581
u	537
click	533
make	527

Table 2. Most common words in non-spam email

Words	Frequency
enron	13382
ect	11417
vince	8531
one	5737
hou	5569
two thousand	4935
kaminski	4770
com	4444
two	4389
please	4386
would	4151
cc	3889
zero	3587
three	3368
pm	3262
two thousand and one	3060
four	2906
research	2676
thanks	2523
time	2466

We can appreciate an unbalance in the data, there are much more non-spam emails than spam emails. In terms of unique words, the raw number in the two categories are almost the same, but taking into account the significant higher number of non-spam emails one can say that average number of unique words in spam emails is actually higher in comparison to non-spam emails. Besides, the number of unique words across all emails does not have a significant overlap, which means that the unique words in spam emails and the unique words in non spam emails are usually different.

The most common words also have a much lower frequency in spam emails even taking into account the ratio of spam emails and non-spam emails. Spam emails also seem to frequently lead to websites ("com", "http", "click"), and offer things or refer to money ("free", "money", "get").

2 PRE-PROCESSING

The process we follow for pre-processing the data is as follows:

- Tokenizing and lowercasing the email texts
- Removing stop words, punctuation and single letters
- Removing "subjects:" or "subject", as this is standard in all emails and would not help us later
- We also converted numbers to their word equivalents, in order to not have repetitions (5 and "five", for example)
- We applied lemmatization to the words, using the NLTK library for our input data for the models. Nevertheless, we tried using the spacy library first and concluded that the added time to lemmatize the words was not worth it compared to the minimal difference in accuracy between the nltk lemmatization and the spacy lemmatization in this case.

3 TEXT CLASSIFICATION

After getting the preprocessed data to a dataframe, we started by splitting the data into the test and the training set. We used 20% for the test set and 80% for the training set. For that, we used the train_test_split function of the scikit-learn package.

3.1 Naive Bayes

We trained a naïve Bayes model using the training set. We used the MultinomialNB classifier of the naïve Bayes part of scikit-learn. We used the test set to find out about the global performance of the model. We used the accuracy_score function of the metrics of scikit-learn.
We tried different vectorization models. With the count vectorizer (Bag of Word), the performance was of about 98.6% of accuracy. With TF-IDF, the accuracy score was of about 88.9%, which is much lower. The last vectorization model that we tried was count vectorizer (bag of 2-gram), and that yielded an accuracy of about 97.3%. So overall, the vectorization model with the best accuracy was the count vectorizer.

3.2 Feed forward neural network

Then, we trained a feed forward neural network. We also tested different vectorization methods. Here are our results :

- For the count vectorizer, the F1 score on the test data was of 97.92% and the accuracy of 99.04%.
- For the TF-IDF, the F1 score on the test data was of 98.69% and the accuracy of 99.30%.
- For the 2-gram, the F1 score on the test data was of 94.12% and the accuracy of 97.82%.
- For the 3-gram, the F1 score on the test data was of 91.35% and the accuracy of 96.51%.
- For the 4-gram, the F1 score on the test data was of 76.58% and the accuracy of 91.15%.

With the Keras library we used the following hyperparameters to build the feed forward neural network with a total of 276'551 trainable parameters:

- It's a sequential model since it only takes one input (list of sentences) and produce one output (list of predictions of the spam field).
- One input layer with input shape 27653 too and output shape (None, 10). It has 276'540 parameters.
- One hidden layer with activation function sigmoid. It has 11 parameters.

3.3 Result Analysis

As expected the TF-IDF vectorization model provides the most accurate results in neural network model, since it normalizes the Bag of Words model. We can see that the results of the naive Bayes model are similar to the ones provided by the feed forward neural network in the best case and has the advantage of being faster, because it does not have to be trained beforehand. It can be said that the Bayes model provides a faster and ready to go approach for classifying spam emails at a minimal cost of accuracy in comparison to the feed forward neural network used.

On the other hand the results of the feed forward neural network shows that there is overfitting, since there is a 100% of accuracy in the training data. Including a drop-out layer to avoid this makes the accuracy of the model lower on the test data, which is why it was not included in the final model.

The bigram model provides a good result in both the naïve Bayes and the neural network model. As the n in the n-gram increases the accuracy falls, which tell us that the amount of data is too low to use this big n.

4 PMI BASED WORD SIMILARITY

For the last task, we converted words to vectors using Word2Vec. We used windows size 3 and trained the model with all the words in the emails. We then chose 20 random words and got the output of the 5 most similar words based on the computed word-word pointwise mutual information matrix. The results can be seen in the tables 3, 4 and 5 at the end of this document.

5 CONCLUSION

The most accurate spam classification model was the feed forward neural network with tf-idf. Nevertheless it takes about five minutes to train while Naive Bayes can directly have an also very accurate result with the simple bag-of-words model. However we have to take into account the fine tuning that can be performed on the model and the average training times of these kind of models. Without taking into account the training time we can reach more than 99 % of accuracy very fast.

Additionally, PMI works very well for finding common words, as we can see in Table 3, 4 and 5. Normally names are grouped with names, numbers with numbers and so on. We have some letters in our final results because even if we got rid of them at the preprocessing part, because sometimes the lemmatizer of NLTK converts some very short words to letter such as "us" to "u".

Table 3. Words and their most similar words

Words	Most similar
	sixteen (82.5%)
	fourteen (82.1%)
	seventeen (81.6%)
	twenty-nine (81.4%)
two	one (81.2%)
	went (96.6%)
	promised (95.5%)
	crazy (96.6%)
	handed (95.2%)
saying	wonder (95.1%)
	five (85.3%)
	three (84.1%)
	fifteen (81.8%)
	six (81.4%)
one	two (81.2%)
	synm (89.4%)
	bardziej (89.2%)
	brett (88.8%)
	ranch (88.8%)
bruzzese	etapie (88.6%)
	graduate (91.0%)
	univ (90.3%)
	harvard (89.8%)
	college (89.3%)
school	sloan (88.0%)
	thank (78.6%)
	shirleys (69.3%)
	summoned (65.6%)
	christie (65.3%)
thanks	yes (64.9%)
	azminab (75.0%)
	baldock (73.3%)
	vkamins (72.4%)
	raul (72.1%)
j	giovannini (71.2%)
	universite (92.6%)
	angela (92.1%)
	dennis (92.0%)
	andrew (91.6%)
dorsey	deng (89.8%)

Table 4. Words and their most similar words

Words	Most similar
	ronnie (92.6%)
	brian (92.4%)
	roman (91.2%)
	ron (91.2%)
elizabeth	li (91.2%)
	wincenty (77.1%)
	homeowner (76.9%)
	vincent (71.4%)
	dr (68.2%)
vince	sir (67.6%)
	five (89.9%)
	six (89.7%)
	seven (89.5%)
	twelve (88.2%)
four	thirty-six (88.1%)
	foward (87.7%)
	hood (84.8%)
	dribbling (82.4%)
	messy (82.4%)
forward	desparately (78.6%)
	erhc (98.5%)
	coupled (98.2%)
	newly (98.0%)
	personne (98.0%)
intermediary	pk (98.0%)
	uecker (93.6%)
	studied (92.9%)
	prashant (92.9%)
	wil (92.8%)
jacobsen	yale (92.8%)
	marketplace (90.6%)
	commerce (90.5%)
	nsurance (90.0%)
	workflow (89.2%)
procurement	breathtaking (89.0%)
	included (88.3%)
	updated (87.6%)
	detailed (87.5%)
	final (87.1%)
summary	written (86.9%)

Table 5. Words and their most similar words

Words	Most similar
	easily (95.1%)
	completely (94.2%)
	seen (94.0%)
	obligation (93.9%)
apply	acceptable (93.9%)
	probationary (87.3%)
	objection (86.9%)
	treat (86.5%)
	remitted (86.1%)
consult	decline (85.2%)
	poszlismy (68.6%)
	supatgiat (68.1%)
	vialogy (67.4%)
	lawson (67.3%)
enron	seminarium (67.1%)
	tomorrow (68.9%)
	place (68.5%)
	flight (67.9%)
	either (67.5%)
time	took (67.5%)