Faculty of Engineering & Technology

Electrical & Computer Engineering Department

Artificial Intelligence - ENCE434

Automatic Spam Review Detection

- **Prepared by :**Rami Zayed 1182030 & Sally Ayoub 1181172
- **Instructor:** Dr. Aziz Qaroush
- **Section:**2

- **Inroduction :**

It has become a common practice for people to read online opinions/reviews for different purposes. For example, if one wants to buy a product, one typically goes to a review site (e.g., amazon.com) to read some reviews of the product. If most reviews are positive, one is likely to buy the product. If most reviews are negative, one will almost certainly not buy it. Positive opinions can result in significant financial gains and/or fames for businesses, organizations and individuals. This, unfortunately, gives strong incentives for opinion spamming.

Opinion Spamming: It refers to "illegal" activities (e.g., writing fake reviews, also called shilling) that try to mislead readers or automated opinion mining and sentiment analysis systems by giving undeserving positive opinions to some target entities in order to promote the entities and/or by giving false negative opinions to some other entities in order to damage their reputations. Opinion spam has many forms, e.g., fake reviews (also called bogus reviews), fake comments, fake blogs, fake social network postings, deceptions, and deceptive messages.

We believe that as opinions on the Web are increasingly used in practice by consumers, organizations, and businesses for their decision making, opinion spamming will get worse and also more sophisticated. Detecting spam reviews or opinions will become more and more critical and not easy.

- # Problem formalization:

In this Project we were given a Raw Dataset from the yelp website , and we were asked to Train a Spam Reviews Detection System ,and To test the system with a new "Not seen before " Data to see the efficiency of this system.

First we balanced the test data knowing that we have a lot more ham than spam,so we took 10000 sample's for each the spam and the ham of a total of 20,000 sample for the test/training data.

we started the design of our system by choosing the features we think would help the most in constructing an efficient Spam detection System , the features were chosen based on reading multi searching Papers regarding the spam review detection systems and the analysis of how the yelp filter might work, we managed to chose the following features :

1-Similarity of sentences.

2- Length of the sentences.

3-Useful count.

4- First Count .

- First : we chose the column "Review Content" in the input Dataset as one of the main structures of a Spam Review, the spammers tend to have a psychology of language , where they use very common used words , they tend some times to generalize their Reviews using non related words , or some times to increase the length of the review they might repeat some words or characters , we were able to extract the Length of the Review content as a Feature and The similarities between sentences as another feature the Article **"What Yelp Fake Review Filter Might Be Doing"** helped a lot in directing us into focusing on the linguistic behavioral of the spammers

The Analysis of the Review Content was done by first , Remove the Stop words which are very common words in the language that won't add up to the value of the review , then we used words steaming and converted everything into a lower case, then we calculated the cosine similarity between all filtered sentences and we got our first feature, then the length of the sentences is computed and we had our second feature .

- Second :
  we chose the third feature which is the first count which is considered one of the spammers abnormal behaviors Like ,viewer id, time of posting, frequency of posting, first reviewers of products, and many more.

  First Count is a good feature since it indicates if the person with a particular IDis always the first person to make a review , since spammers most likely try to be the first to comment or make a review , this could be a hint that thee reviews with this person's ID will most likely be fake , we tried to use the reviewer ID as a feature but It generated a very low quality results so we dropped it .

- Third :
  The last feature we used is the useful count , which is the number of times this review marked as useful , spammers tend to " overreact" into supporting the reviews that agree with their intentions so if a Review is marked very useful or very not useful sometimes it might be a spam.

Output :

we used the naïve bias classifier to classify the output for the Dataset , the naïve bias classifier generates a confusion matrix which has the True classification , worng classifications that were classifies ""True", Wrong classifications and True

classifications classified as “Wrong “ , confusion matrix helps finding the Recall and precision of the System .

the results we got using the naïve bias classifier and the 4 mentioned features are :

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Accuray:  0.5652466745243307
Precision:  0.6207571801566579
Recoil:  0.32215447154471544
[[ 951 2001]
 [ 581 2406]]
['/content/drive/MyDrive/Colab_Notebooks/naiv_1.pkl']
```

Fig-1

we tried using the ReviwerID as another additional Feature but the results we’re less efficient than using 4 features Fig 2 below shows the results we got using 5 features including ReviwerID :

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
Accuray:  0.5648366453351297
Precision:  0.601765571358509
Recoil:  0.40913637879293097
[[1227 1772]
 [ 812 2127]]
```

Fig -2

Another model we used is Feature Extraction using TFIDF tool ,the product of TF in Fig 3 and IDF in Fig -4 will give us a measure of how frequent the word is in a document multiplied by how unique the word is, giving rise to Term Frequency-Inverse Document Frequency(TF-IDF) measure we obtained the results in Fig5 which are good results that depend mainly on the content of the review .

$$Term\ Frequency(t) = \frac{number\ of\ times\ t\ appears\ in\ a\ document}{total\ number\ of\ terms\ in\ the\ document}$$

Fig-3

$$IDF(t) = log_e\left(\frac{Total\ Number\ of\ Documents}{Number\ of\ Documents\ with\ t\ in\ it}\right)$$

Fig -4

```
  warnings.warn(msg, category=FutureWarning)
accuracy:  0.6478333333333334
Precision:  0.6565860215053764
Recall:  0.6417077175697865
              precision    recall  f1-score   support

         0.0       0.66      0.64      0.65      3045
         1.0       0.64      0.65      0.65      2955

    accuracy                           0.65      6000
   macro avg       0.65      0.65      0.65      6000
weighted avg       0.65      0.65      0.65      6000
```

Fig-5