CLASSIFICATION OF EMAILS

M. Sai Nikhil (130050055)

P. Amar Sekhar (130050084)

Ashish Chauhan (130050087)

Project Description:

Emails are an integral part of our internet life. Often we suffer from the lack of classification of the emails we received. So many people find it tiresome to search for a particular type of email from a large heap of them. So we, through this project aim to develop a machine learning algorithm which classifies the emails received using a train data set.

Implementation:

The task is to distinguish between two types of emails, *"spam"* and "non-spam" often called *"ham"*. The machine learning classifier will detect that an email is spam if it is characterised by certain features.(eg – words like "lottery" ,"offers", "Join now!"– is crucial words in spam detection and offers some of the strongest clues:)

1) Pre-Processing Data**:**

- We will be using the Python-based library Natural Language Toolkit (**NLTK**), which has rich functionality in natural language processing tasks.
- Splitting the text by white spaces and punctuation marks – the tools that are used for this purpose are called ***tokenizers**,* and you can use a tokenizer provided with the NLTK.
- Linking the different forms of the same word (*price* and *prices, is* and *are*) to each other – the tools that can do that are called ***lemmatizers**,* and you can again use one of those that come with the NLTK.
- converting all words to lowercase so that the classifier does not treat *People, people* and *PEOPLE* as three separate features.
- Some words like "*the*", "*is*" or *"of"* appear in all emails, don't have much content to them and are therefore not going to help you distinguish spam from ham. Such words are called ***stopwords*** and they can be disregarded

during classification. NLTK has a corpus of **stopwords** for several languages including English.

2) **Feature Extraction:**

Features - Words can tell the program whether the email is spam or ham .

For each word that is not in the stopword list calculate how frequently it occurs in the text.This approach is called the *bag-of-words (bow,* for short*)*, and it allows the classifier to notice that certain keywords may occur in both types of emails but with different frequencies (for example the word “offer” is much more frequent in spam than ham emails).

This way we can extract the features from the emails and pair them with the email class label (“spam” or “ham”).

3) ***Training a classifier :***

Now that the data is in the correct format, we split it into a training set that will be used to train the classifier, and a test set that will be used to evaluate it.

We applied classifier **Naive Bayes classifier**, which is a simple yet powerful classification algorithm.

The classifier tries to choose the most probable class, or label, among the two classes, spam and ham, i.e. $c \in \{spam, ham\}$ based on what it has learned about the features (presence or frequency of words in the emails of each type). More precisely, it’s trying to choose the most probable class given the words in the e-mail:

$$\hat{c} = argmax_{c \in \{spam, ham\}} P(c|words)$$

DataSet:

We use a large corpus of real world email messages from Enron employees. The Enron corpus was made public during the legal investigation concerning the

Enron Corporation. This dataset, along with a thorough explanation of its origin, is available at http://www-2.cs.cmu.edu/~enron/. In the raw Enron corpus, there are a total of 619,446 messages belonging to 158 users.

Research Papers:

[1] Bryan Klimt, Yiming Yang. Introducing the Enron Corpus.

[2] Ron Bekkerman. Automatic Categorization of Email into Folders: Benchmark Experiments on Enron and SRI Corpora. 2004.

[3] Wikipedia:Hierarchical clustering.
https://en.wikipedia.org/wiki/Hierarchical_clustering

[4] Emails classification by data mining techniques.

Mohammed A.Naser, Athar H.Mohammed

Department of Computer, College of Sciences for Women, University of Babylon