Assignment 4: Neural Networks (Text Classification)

Natasa Farmaki - DS3517018

Dimitris Georgiou - DS3517004

Stratos Gounidellis - DS3517005

Course: Text Engineering & Analytics

Professor: Ion Androutsopoulos

Athens, March 2018

INTRODUCTION

The aim of this assignment is to classify a set of email as legitimate or spam. It is a fact that unsolicited commercial or bulk e-mail also known as spam constitutes a big trouble over the internet. Spam are considered waste of time, storage space and communication bandwidth. Thus, spam filtering is way to ameliorate the situation. More specifically, spam filtering and detection is the process of detecting spam emails and prevents those messages from getting to a user's inbox.

In our approach, we use three different methods to classify the emails in the two categories, i.e. legitimate and spam emails. To be more specific we applied a baseline classifier, Multilayer Perceptron (MLPs) and Convolutional Neural Networks (CNNs). In order to evaluate their effectiveness, we utilized different metrics, such as accuracy, recall, precision and F1-score because the dataset is imbalanced as the spam emails are almost the half of the legitimate emails. In addition, we compared MLPs and CNNs approaches with a simplistic baseline approach and we generated learning curves and precision-recall curves.

The python code is available in the following link: https://thinkingtea.github.io/repoTEA/

DATA DESCRIPTION

The Enron Corpus is a large database of over 600,000 emails generated by 158 employees of the Enron Corporation and acquired by the Federal Energy Regulatory Commission during its investigation after the company's collapse. This dataset is quite interesting as it is one of the few large email corpus available in the public domain.

The Enron data was originally collected at Enron Corporation headquarters in Houston during two weeks in May 2002 by Joe Bartling, a litigation support and data analysis contractor working for Aspen Systems, whom the Federal Energy Regulatory Commission (FERC) had hired to preserve and collect the vast amounts of data in the wake of the Enron Bankruptcy in December 2001.

In the current subset, we used a subset of the above dataset with 5172 pre-processed emails (3672 legitimate emails and 1500 spam emails). That subset is available here.

DATA PRE-PROCCESING

For this assignment we chose to preprocess the data in two different ways. We use both TF-IDF vectors and simple tokens. The reason is that CNNs are fed with simple tokens and we want to compare MLPs performance with CNNs. Also, we want to compare MLP's performance with the performance of the classifiers of assignment two and that is why we also train MLPs with TF-IDF vectors. To be more specific after obtaining the data, we split the emails in a training and test set, with the latter equal to the thirty percent (30%) of the initial dataset (3620 training data and 1552 test data). Additionally, the label is encoded with 0 and 1, with 0 standing for "legitimate" and 1 standing for "spam". Then, we convert the collection of the emails to a sparse matrix of TF-IDF features that we will use to feed the first edition of MLP. We use unigrams, bigrams and trigrams.

When building the vocabulary, we chose to ignore terms that have a document frequency strictly lower than a given threshold, which in our case is set to ten (10). This value is also called cut-off in the literature. In that way, we remove the insignificant words, which appear in the sentences and do not have any meaning or indications about the content. Finally, we normalize the features to be between -1 and 1.

For the second edition of MLP and the CNN, as mentioned above, we also tokenize the emails in order to feed both MLP and CNN and keep the most common words. Secondly, we turn the input into numerical arrays. Finally, pad sequences is used to ensure that all sequences in the list have the same length. Overall, we have two different data set, one for the first MLP and one for the second MLP and the CNN.

POS TAGGERS

As mentioned above we will use 3 classifiers to find the corresponding tags for each word. The first is the simplest Baseline Classifier and 2 Neural Networks classifiers (MLP, CNN)

Baseline Classifier
As stated above, the dataset is imbalanced, which means that the legitimate messages are far more than the spam messages. Therefore, it is useful to construct a baseline classifier that would just classify all emails as legitimate. In other words, in the baseline algorithm, legitimate messages are (correctly) never blocked and spam messages (mistakenly) always pass.

Neural Networks
Neural Networks are essentially mathematical models to solve an optimization problem. They are made of neurons, the basic computation unit of neural networks. A neuron takes an input (say x), do some computation on It (say: multiply it with a variable w and adds another variable b) to produce a value (say; z= wx+b). This value is passed to a non-linear function called activation function (f) to produce the final output (activation) of a neuron. There are many kinds of activation functions. One of the popular activation function is Sigmoid, which is Y = 1/(1 + e^(-z)).

Multilayer Perception(MLP)
The Multilayer Perception (MLP) is perhaps the most popular network architecture in use today both for classification and regression. MLPs are feed forward neural networks which are typically composed of several layers of nodes with unidirectional connections, often trained by back propagation. The learning process of MLP network is based on the data samples composed of the N-dimensional input vector x and the M-dimensional desired output vector d, called destination. By processing the input vector x, the MLP produces the output signal vector y(x,w) where w is the vector of adapted weights. The error signal produced actuates a control mechanism of the learning algorithm. The corrective adjustments are designed to make the output signal y_k (k = 1, 2,...,M) to the desired response d_k in a step by step manner. The learning algorithm of MLP is based on the minimization of the error function defined on the learning set (x_i,d_i) for i = 1, 2,...,N using the Euclidean norm.

Convolutional neural network (CNN)
Convolutional neural network (CNN) is a class of deep, feed-forward artificial neural networks. CNNs use a variation of multilayer perceptrons designed to require minimal preprocessing. They consist of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers. Convolutions are used over the input layer to compute the output. This results in local connections, where each region of the input is connected to a neuron in the output. Each layer applies different filters and combines their results.

EVALUATION METRICS

In order to evaluate the results of the multiple classifiers and therefore compare their performance we need to implement appropriate measures. The metrics that can holistically evaluate performance are accuracy, recall, precision and F1-score(produced by both recall and precision). In this problem except from comparing the classifiers we are interested in deciding whether collecting more training

instances will improve performance of the classifiers. For that reason, we also construct learning curves based on accuracy and on F1-score.

Accuracy is the number of correct predictions made divided by the total number of predictions made, multiplied by 100 to turn it into a percentage.

Precision is the number of True Positives divided by the number of True Positives and False Positives. Precision can be thought of as a measure of a classifier's exactness. A low precision can also indicate a large number of False Positives.

F1-score is the 2*((precision*recall)/(precision + recall)). Put another way, the F1 score conveys the balance between the precision and the recall.

Recall is the number of True Positives divided by the number of True Positives and the number of False Negatives. Put another way it is the number of positive predictions divided by the number of positive class values in the test data. It is also called Sensitivity or the True Positive Rate. Recall can be thought of as a measure of a classifiers completeness. A low recall indicates many False Negatives.

Learning Curve is the representation in graph form of the rate of learning something over time or repeated experiences. In a text classification problem, a learning curve shows whether collecting more training instances will improve the performance of the classifier both in training and test set. It is important to note that learning curve is not useful for model assessment.

OTHER GRAPHICAL TOOLS

Apart from these metrics we also construct graphical tools to demonstrate the results. More specifically we construct a confusion matric and a precision-recall curve.

Confusion Matrix:
In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa).

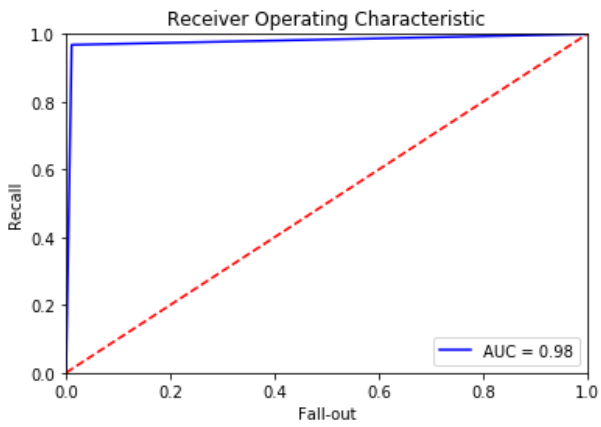
Plot precision-recall curve:
Precision-recall curve is a way to observe the trade-off between precision and recall, and the balance between them. Depending on the requirement (high precision at the cost of recall, or high recall with lower precision), an appropriate algorithm can be chosen.

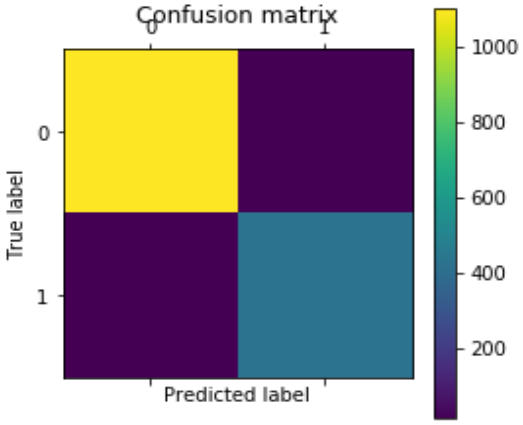
EXPERIMENTAL RESULTS

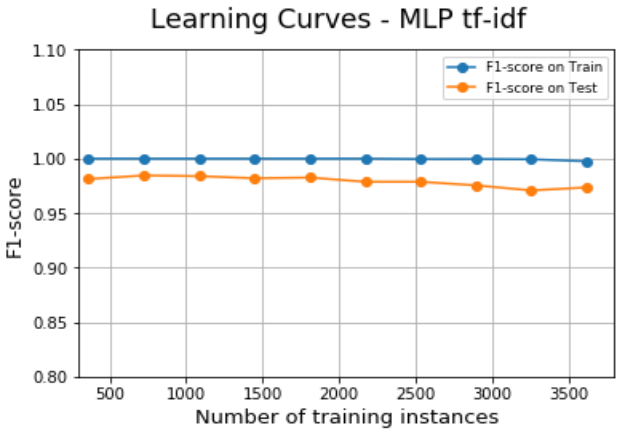
MLP USING TFIDF

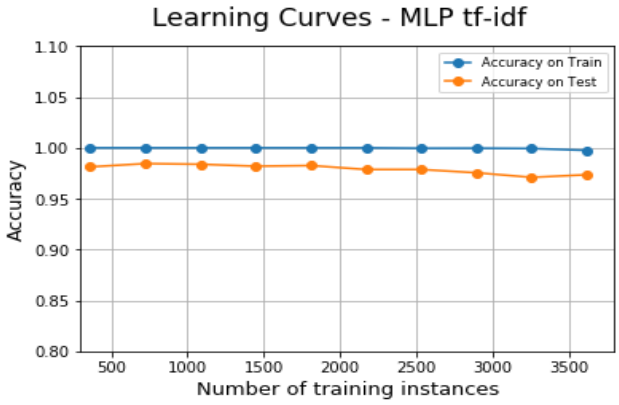
In order to tune the current MLP classifier parameters, we have ran multiple experiments in order to choose the most efficient number of layers, nodes and epochs.

Test Accuracy: 0.98324
Test Precision: 0.97038
Test Recall: 0.97038
Test F1-score: 0.97038
Total cost ratio - MLP TF-IDF vs. baseline classifier: 11.552631578947368







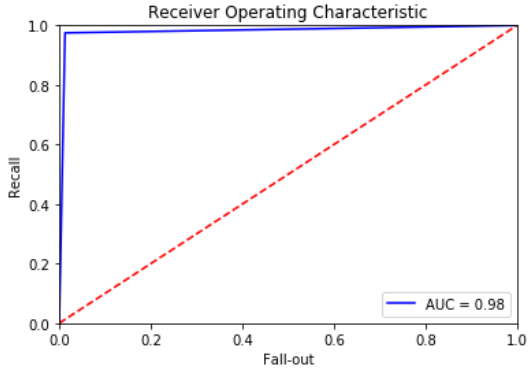


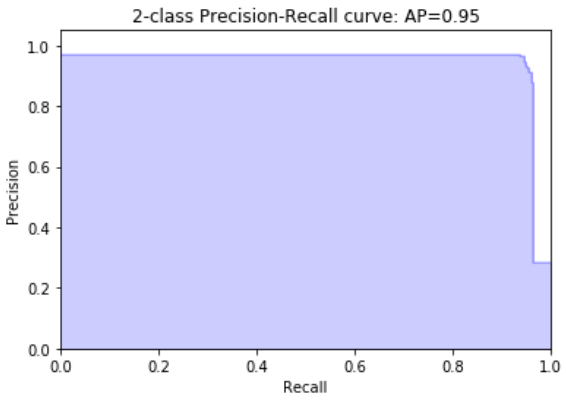
MLP USING TOKENS

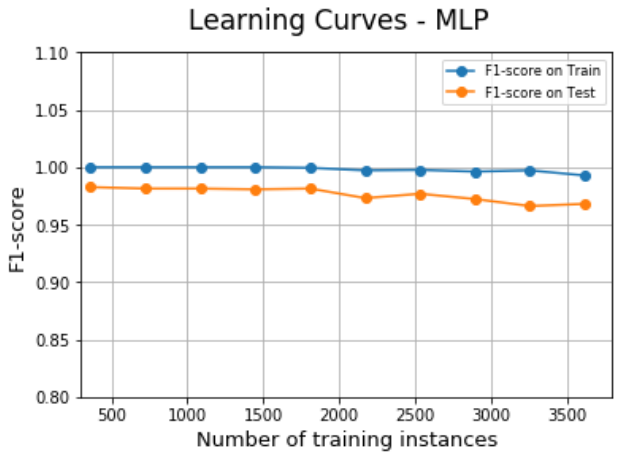
In order to tune the current MLP classifier parameters, we have ran multiple experiments in order to choose the most efficient number of layers, nodes and epochs.

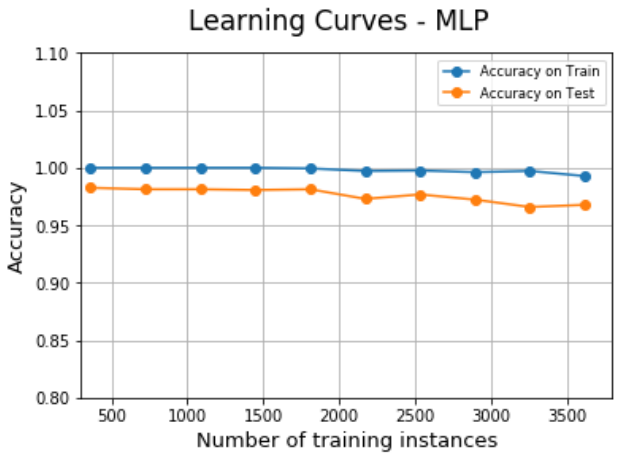
Test Accuracy: 0.95296
Test Precision: 0.85882
Test Recall: 0.997722
Test F1-score: 0.92307
Total cost ratio - MLP vs. baseline classifier: 11.256410256410255

The following graphs show the learning curves with the average F1-score, precision, recall and accuracy for the MLP classifier.







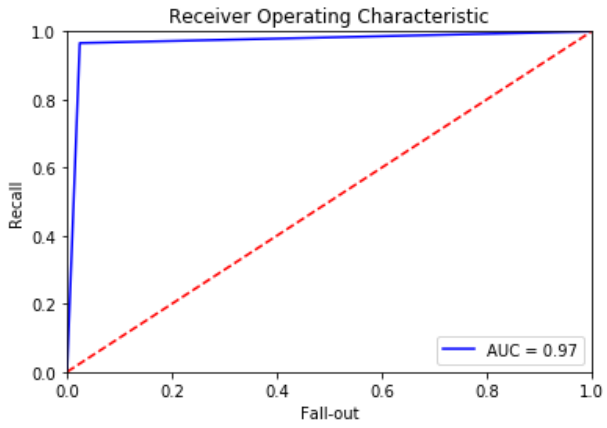


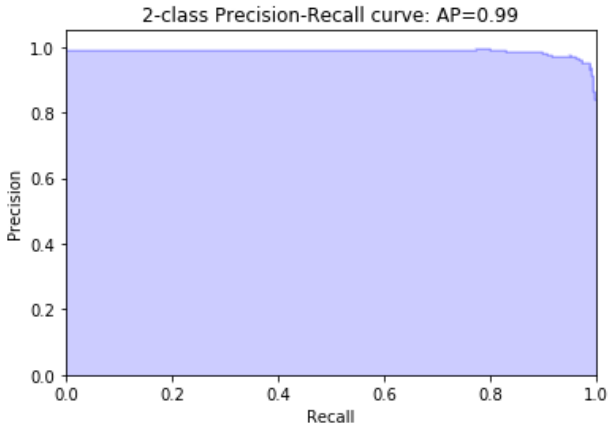
CNN USING TOKENS

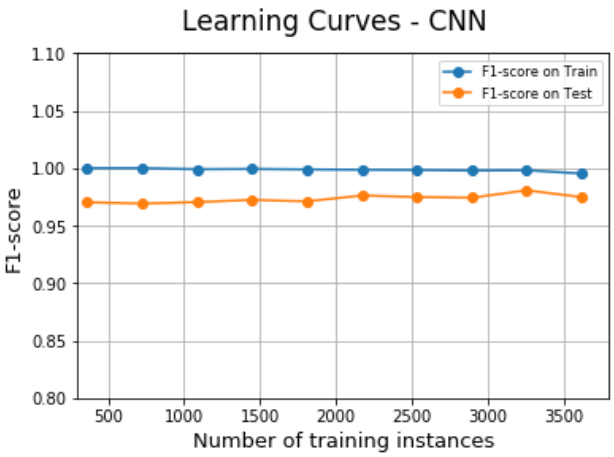
In order to tune the current CNN classifier parameters, we have ran multiple experiments in order to choose the most efficient number of layers, nodes and epochs.

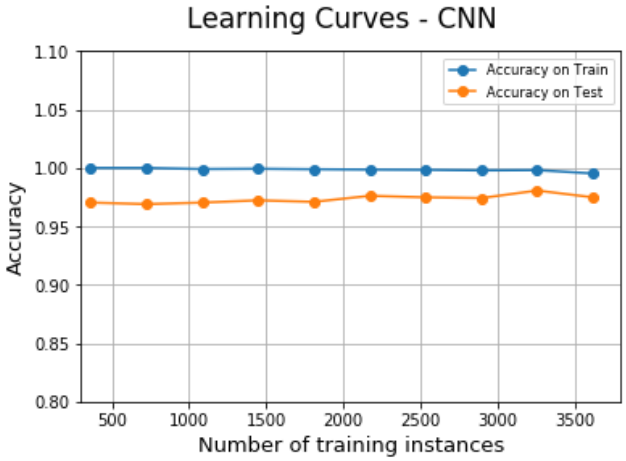
Test Accuracy: 0.97293
Test Precision: 0.94606
Test Recall: 0.95899
Test F1-score: 0.95248
Total cost ratio - CNN vs. baseline classifier: 6.36231884057971

The following graphs show the learning curves with the average F1-score, precision, recall and accuracy for the CNN classifier.









CONCLUSIONS

While comparing the baseline approach with the aforementioned neural network approaches it is obvious that the latter approaches outperform. More specifically, the Multilayer Perceptron using as features the tfidf sparse matrix performs metrics as far as accuracy, F1-score and precision is concerned as we can conclude from the following table. Additionally, the Multilayer Perceptron using as features the tfidf sparse matrix presents the highest total cost ratio versus the baseline approach as shown above.

Results Summary Table

	BASELINE	MLP - TFIDF	MLP - TOKENS	CNN - TOKENS
F1-score	-	0.97038	0.9230769	0.95248
Accuracy	0.83527	0.98324	0.9529639	0.97293
Recall	-	0.97038	0.9977221	0.95899
Precision	-	0.97038	0.8588235	0.94606