UNIVERSITY OF ATHENS
DEPARTMENT OF INFORMATICS AND TELECOMMUNICATIONS

Deep Learning for NLP

Student name: *Lampropoulos Konstantinos*
sdi: 1800092

Course: *Artificial Intelligence II (M138, M226, M262, M325)*
Semester: *Fall Semester 2023*

Contents

1. Abstract

In this task,we were asked to classify some greek tweets based on their emotion in 3 categories: positive, neutral, negative. The tweets contain political views of some people ,who either express their positivity towards a political party or their negativity.There are some that mostly provide facts and do not show emotion,which are categorized as neutral.Natural Language Processing is a difficult task which has many prerequisites before the classification takes place.To begin with,we need to pre-process our data,so irrelevant words are dismissed.Some of those words are: stopwords, non-greek words,retweets,links etc.The pre-processing method helps us decrease the size of our texts (so less computing power is required) and make it easier for the classifier to identify relations between the words and the sentiment provided by them.After pre-processing we are required to tune the Logistic Regressor classifier,using the train and validation sets.Finally we will try to predict the labels for the test set and save them in a .csv file.

2. Data processing and analysis

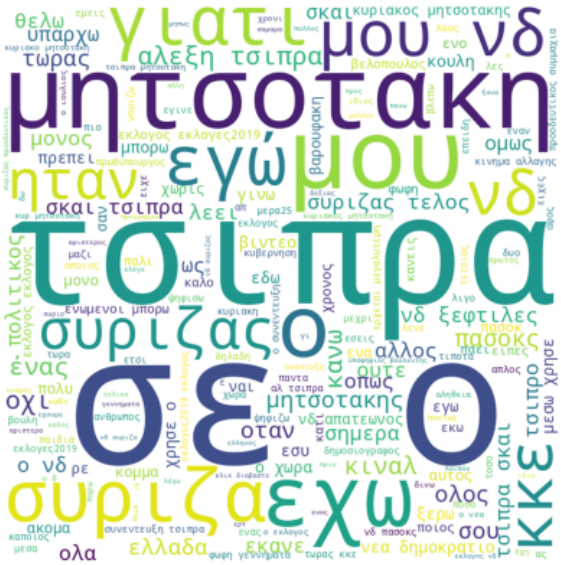
2.1. Pre-processing

In this section we will cover the pre-processing techniques involved.To begin with,we drop the unnecessary columns(NewID , Party) since we will not be using them in the classification process.After that, we convert all the text column in lowercase letters,remove the accents ,as well as, the stopwords and replace them with a single whitespace character.Following that,we replace multiple whitespace characters with a single white space,remove any links and non-greek words as well as retweets (and replace them with single whitespace characters).Finally,we lemmatize the text column to replace words with their dictionary values.The purpose of this pre-processing action is to breakdown the text and remove any unnecessary words and also provide the classifier with the dictionary value of the words so they are more easily recognized.

2.2. Analysis

2.2.1. Word Cloud.

In a word cloud,the most common words have greater size compared to the rest.Below I showcase an image with the results of the wordcloud implementation:



As we can see ,words like: Μητσοτάκη,Τσίπρα,νδ,συριζα, are really common in the database.We also have some words like: ο,σε,εχω ,which shouldn't be kept after removing stopwords.This is the resulting wordcloud after including manually more stopwords to be removed:

It seems like now the most common words that appear are ones that are of meaning in our current project.

2.2.2. Token Frequency.

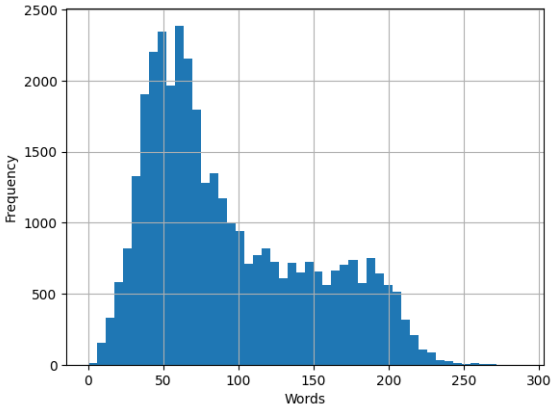
Token frequency is used to find the most common words in the dataframe.In this dataframe our 10 most common words are:

word	count
τσιπρα	9690
νδ	8170
μητσοταχη	7005
εκλογος	4325
συριζας	4189
χχε	3525
συριζα	3401
2019	3164
σχαι	2943
μητσοταχης	2254

We can see that the year 2019 appears,but this poses the question as to if dates and numbers should be removed or not.I personally believe that they should remain ,because they may represent years where some political parties were governing and the debts associated with that period.

2.2.3. Text Length Distribution.

Text length distribution is a method which finds the most common text lengths in the dataframe.



As we can see,the most common text lengths are between 35 to 75 words,which is the size of a common text or tweet.

2.3. Data partitioning for train, test and validation

The partitioning of the dataset remained the same as the one given at the assigment of the project.

2.4. Vectorization

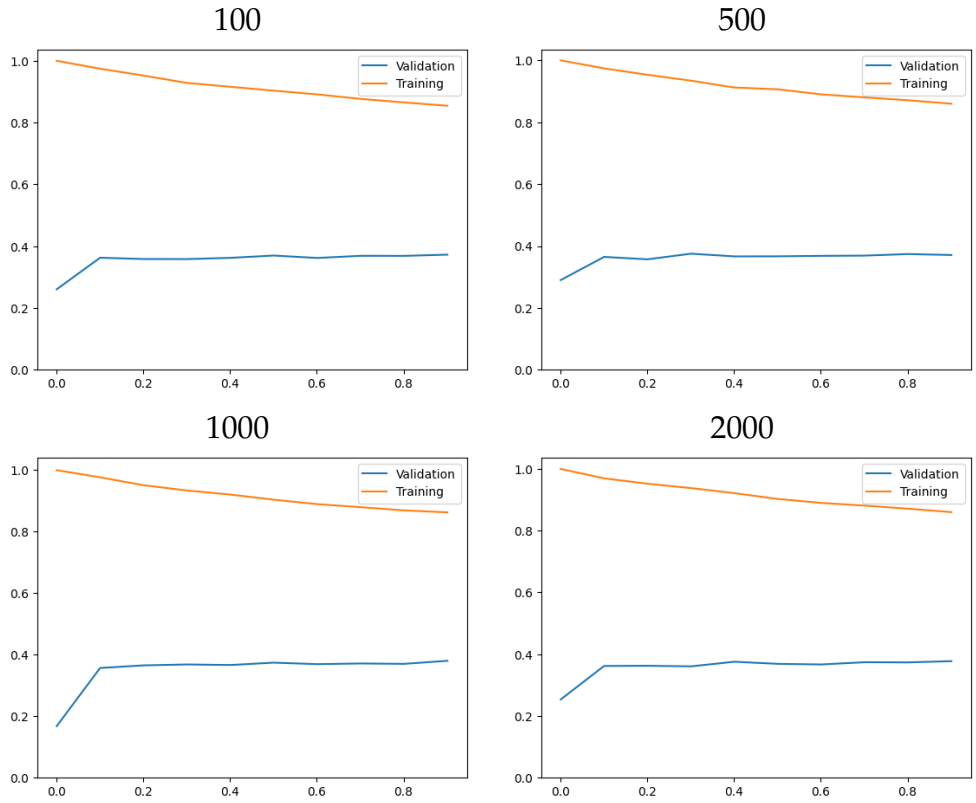
The vectorization technique used was the Count Vectorizer.This vectorizer breaks down the texts of the dataframe in words,creates a vocabulary with the unique words that appear and find the number of times each one appears(in the dataframe).By implementing that technique the most used words are marked as the most important.In

our case that includes words as: τσιπρα,νδ,μητσοτακη etc.
For the labels,the label encoder was used,which converts the unique label values to unique integer values.

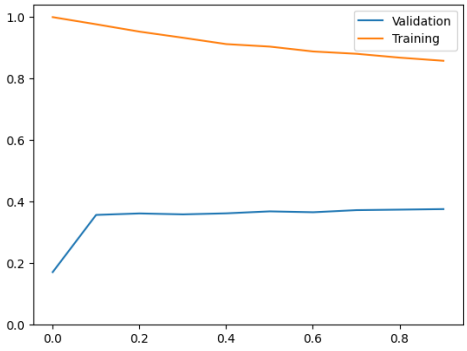
3. Algorithms and Experiments

3.1. Experiments

The first run was a brute force run,with some parameter tuning inspired by the Lab examples(e.g. max iteration count) which yielded really good results as shown at the first row of the table of trials below.Afterwards all the trials are performed at the pre processed dataset.There is no learning curve for this run since it was not part of the cross validation technique implemented below.
The method cross validation with a set number of epochs(10), was used to find the average F1,Precision and Recall scores for each parameter change,as well as the best F1 score from each run.Apart from these constants ,we also need to keep in mind the training f1 scores which,for some cases,will be shown.
The first parameter i experimented with ,was the Maximum number of iterations.I started with 100 iterations and went up to 4000.The best results were for 500 iterations and 2000,were the avg Scores as well as the best F1 were really close.Even though the best F1 score was on the run with 1000 iterations ,the average scores are lower than the previous ones.The learning curves for all the experiments are below:

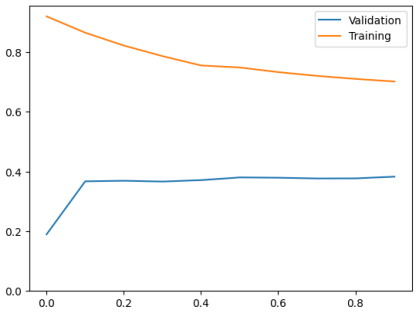


4000

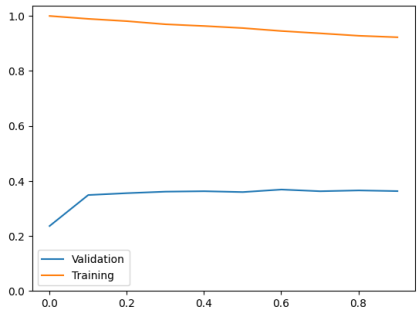


By looking at the learning curves ,the only significant difference is the starting F1 scores,which is why runs with iterations 100,1000,4000 have lower Average Scores.
The next parameter I experimented with,was the C parameter,which is used to prevent overfitting.The default value is 1.0,I tried the values: 0.1,5.0,10.0,20.0,100.0 which created the learning curves below:

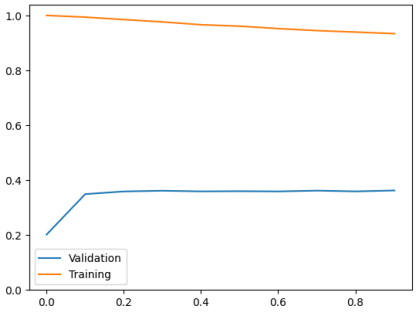
C = 0.1 Iter = 500



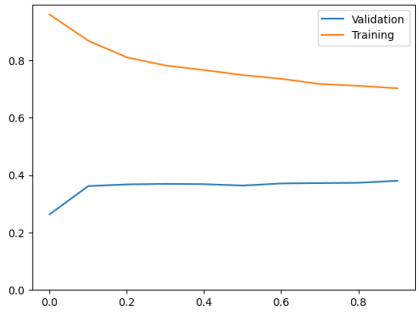
C = 5.0 Iter = 500



C = 5.0 Iter = 2000

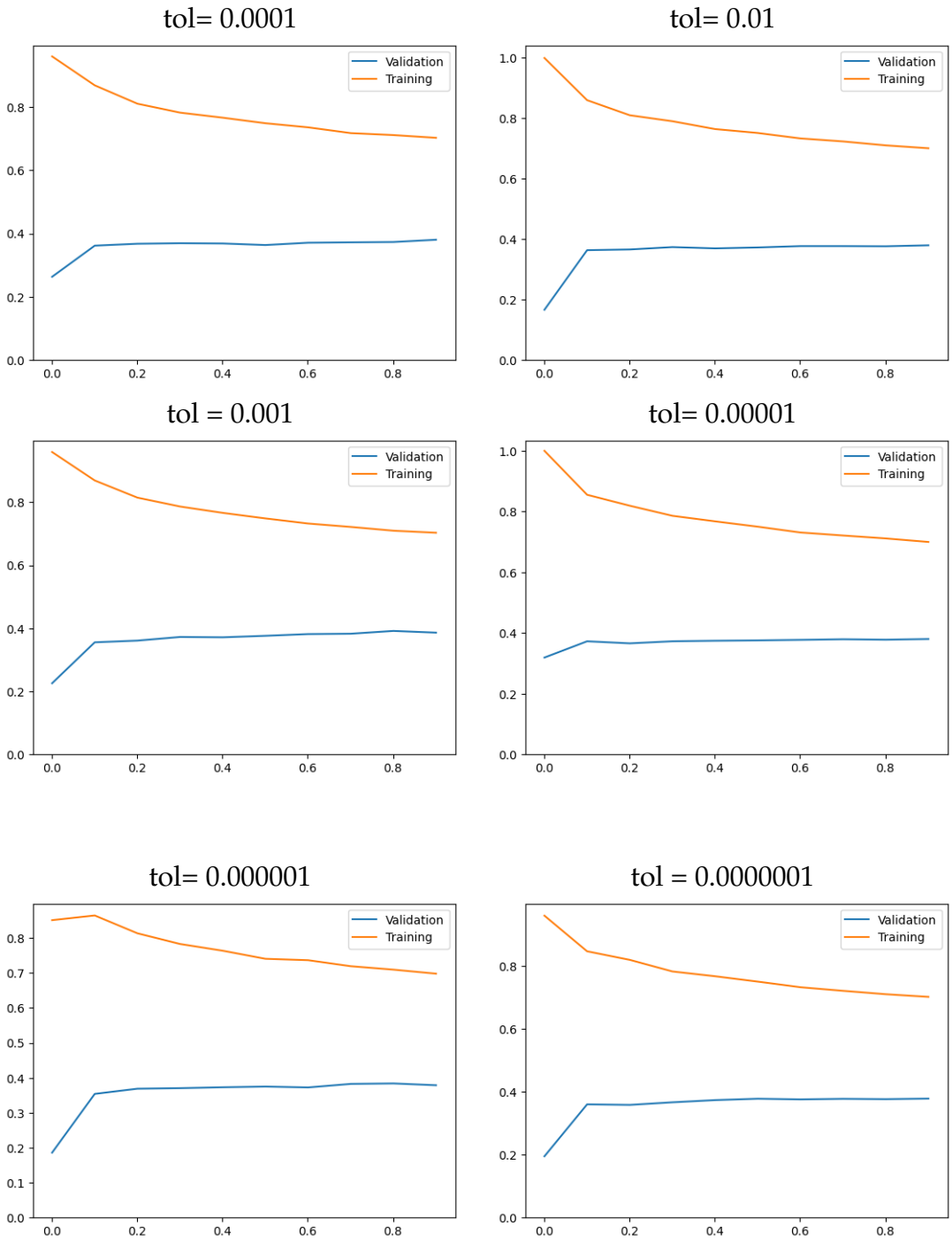


C = 0.1 Iter = 2000



As we can see,by implementing stronger regularization to the model,we get greater validation scores,but lower training scores. Training score was 77.3% ,compared to 90%+ traning scores of the others.When C = 5.0 ,an error from occured which stated that the algorithm did not converge,which meant that 500 iterations were too low for the current tuned model.So the next best number of iterations is 2000 and that is what was used. As regularization value,C = 0.1 was selected as it provides the best model and doesn't showcase any potential overfittings.
The next tuning that took place,was about the tolerance which whether to converge or not.Convergence happens when the log likelihood changes are equal or less than

the given tolerance.As shown below,the best tolerance was found to be 1e-3 (or 0.001) which was the one used afterwards.



As we can see from the learning curves,the closer ones ,are the ones where tolerance is 1e-3.The smoother one is the one with tolerance 1e-6,but since we store the best model and use that for the prediction of the test set,we are more interested in the best f1 score.

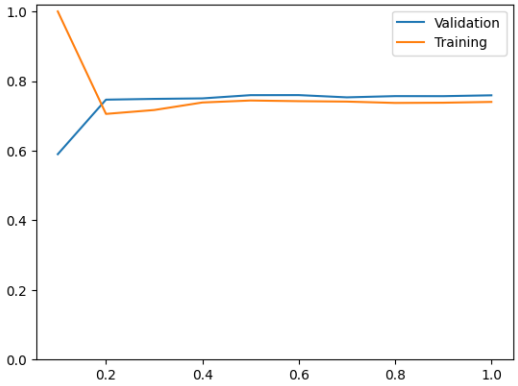
3.1.1. Table of Trials. .

C	Max Iterations	Tolerance	Avg F1	Avg Recall	Avg Precision	Best F1
1.0	2000	0.0001	37.3%	37.3%	37.3%	
1.0	100	0.0001	35.4%	36.2%	36.4%	37.2%
1.0	500	0.0001	36.0%	36.3%	36.3%	37.5%
1.0	1000	0.0001	34.9%	36.5%	35.4%	38.0%
1.0	2000	0.0001	35.7%	36.3%	36.4%	37.7%
1.0	4000	0.0001	34.6%	36.3%	35.8%	37.6%
0.1	500	0.0001	35.7%	37.0%	36.0%	38.2%
5.0	500	0.0001	35.5%	36.1%	36.1%	37.1%
5.0	2000	0.0001	35.5%	36.1%	36.2%	37.4%
10.0	2000	0.0001	34.7%	35.8%	35.7%	36.6%
20.0	2000	0.0001	34.8%	35.8%	35.8%	36.8%
100.0	2000	0.0001	34.3%	35.6%	35.5%	36.2%
0.1	2000	0.0001	35.5%	37.1%	36.4%	38.1%
0.1	2000	0.01	35.2%	36.9%	35.7%	38.0%
0.1	2000	0.001	36.0%	37.1%	37.1%	39.2%
0.1	2000	0.00001	36.9%	37.0%	37.0%	38.0%
0.1	2000	0.000001	35.4%	36.5%	35.6%	37.4%
0.1	2000	0.0000001	35.3%	36.8%	36.6%	37.8%

Table 1: Trials

3.2. Evaluation

The given results showcase under-fitting,since we can clearly see from the learning curves,that despite the tuning and optimization of the classifier, the validation score remains significantly low,compared to the training score.In scenarios where neither under-fitting nor over-fitting take place we would both learning curves be much closer.An example is the learning curve by the lab of the course:

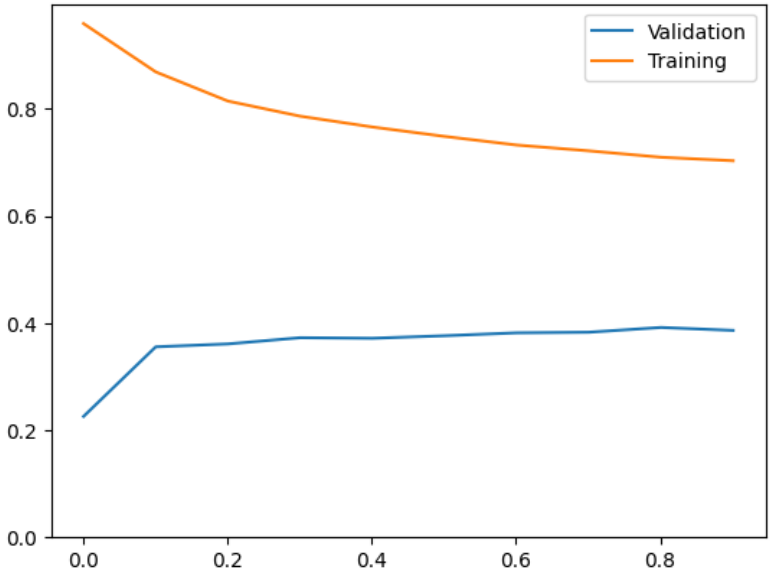


We can clearly see that the model is not under-fitting since the training score are similar to the test scores,and we can also see that the model is not overfitted since it can predict ,at the same percentage of the training set,the test set.Keepin that in mind,we can conclude that the given dataset underfits our model and to yield proper results ,we would need a larger training set.
The metrics used were F1-Score,Precision,Recall.F1-Score is the harmonic mean of Precision and Recall.Precision is the percentage of correctly identified positive values to

the total number of predicted positive values.Recall is the percentage of the correctly identified positive values to the total number of values. F1-Score provides a balance between those 2 metrics and that is why it is the preferred metric to tune and optimize models in machine learning.

3.2.1. Learning Curve.

The learning curve from the best experiment is :



The learning curve shows that even though we pre-processed and tuned the model,it is heavily under-fitted,and would need more tweets in order to be able to correctly predict the sentiment for a higher percentage of tweets.

4. Results and Overall Analysis

NLP (Natural Language Processing) is generally hard ,since it requires from computers to understand our languages which are complex and not always straighforward. Given that,and the fact that the classifier is underfitting(to properly fit the classifier we would need a larger training set),the results are expected.Correctly identifying the sentiment of 39.2% of tweets at average,is low.NLP ,as I stated before,is a hard and complex procedure where classifiers may not be able to provide as good results as Neural Networks (but Neural Networks are highly computing consuming) and require a lot of processing power (usually high end GPUs) to run efficiently.

5. Bibliography

References

[1] More about the Logistic Regression Lab