Language Models!

Summer 2023

Mehrdad Mohammadian

Table of Contents

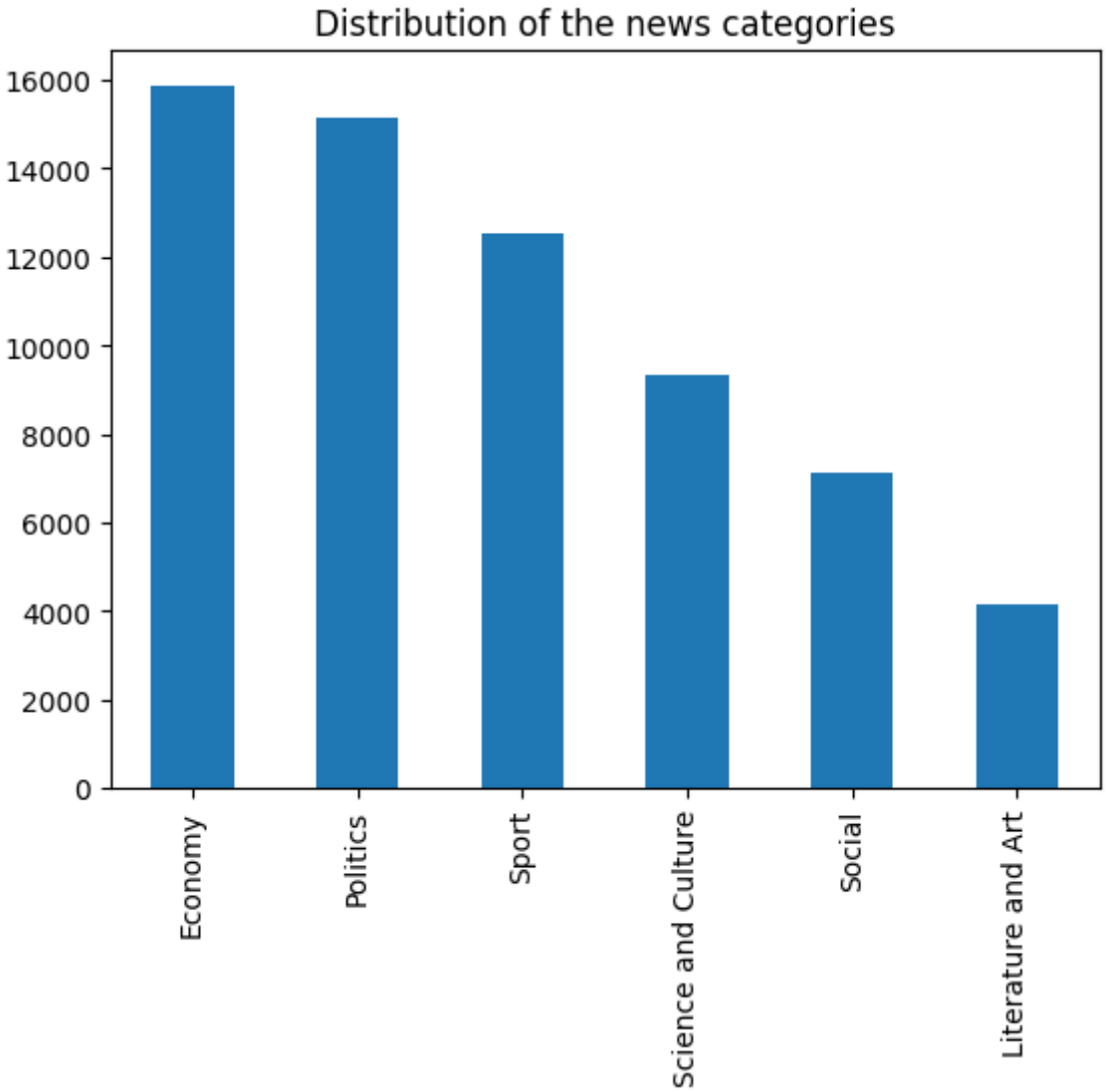
Problem **3**
Dataset **3**
N-gram Models **4**
Normalization 4
Stopwords 5
Unseen N-grams (Smoothing) 7
Modern Models **8**
BERT 8
Feature Extraction 8
Fine-tune 9
Environment **10**

Problem

The purpose is to classify the “Hamshahri News” dataset using N-gram language models and modern language models such as BERT.

Dataset

The training dataset consists of 64140 samples of news from the Hamshari website. The bar chart below shows the distribution of their corresponding categories.

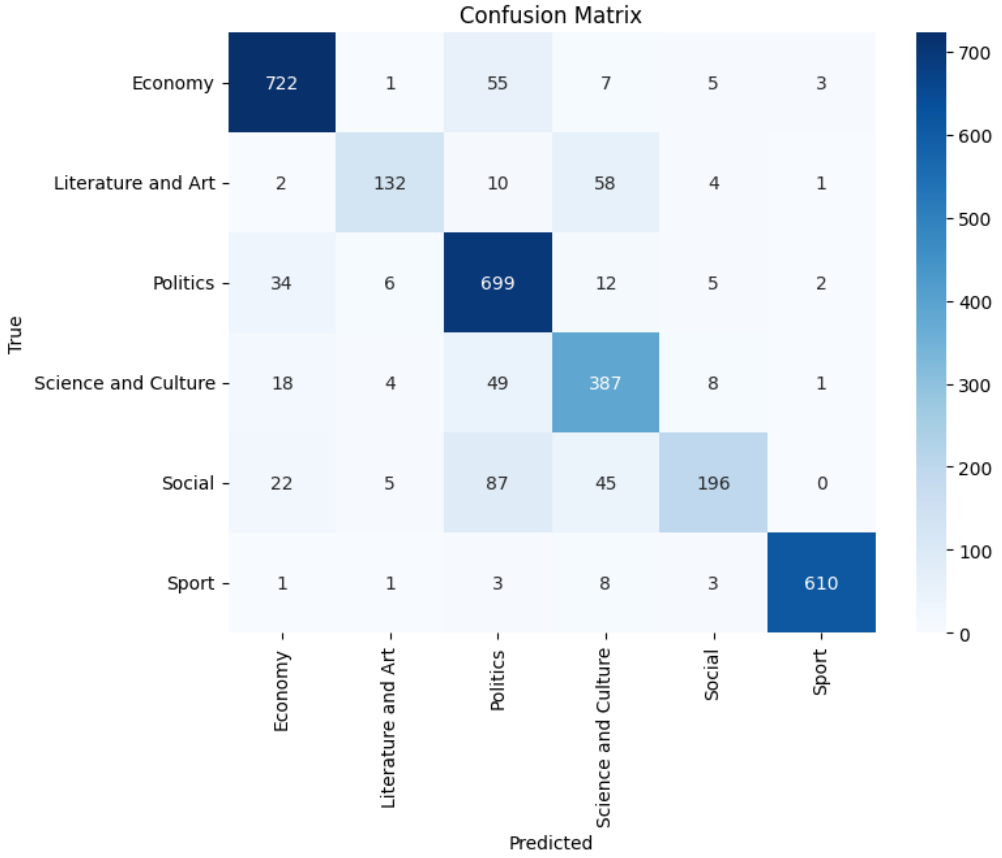


N-gram Models

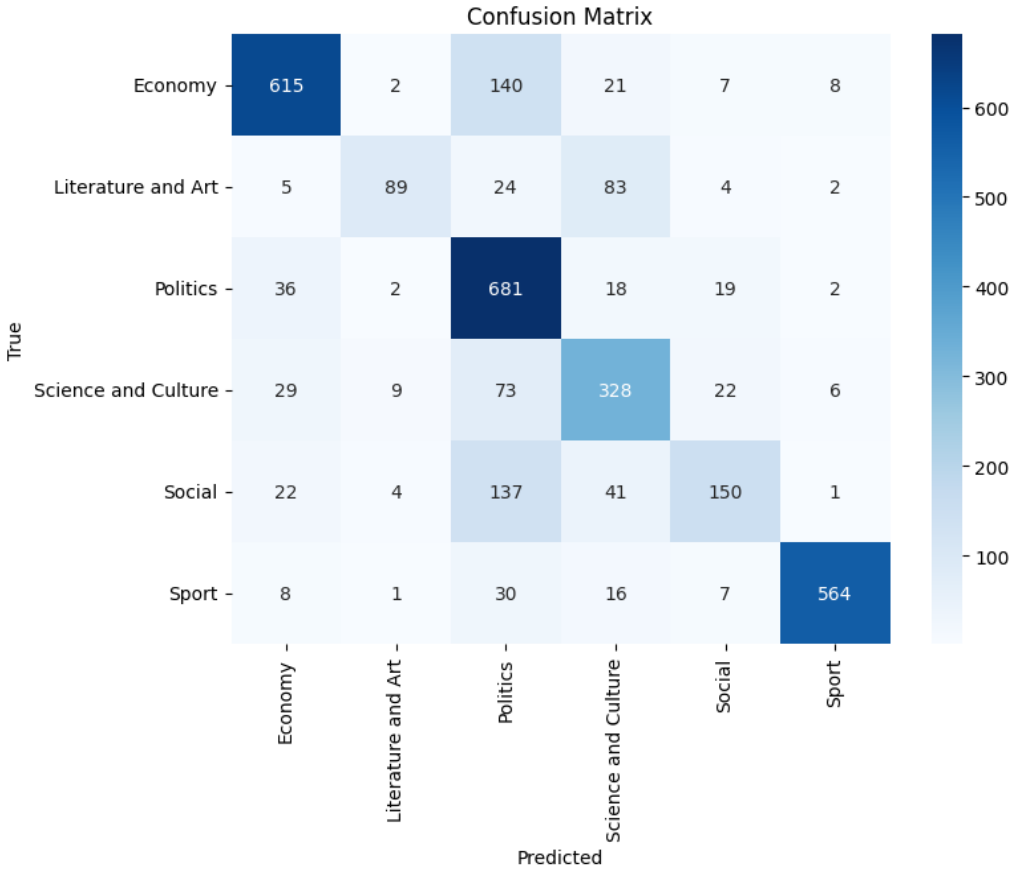
Normalization

I used the Hazm library to normalize the dataset. However, after applying the Hazm normalizer the accuracy drops about 10 percent on the test dataset. May normalization does not work on this dataset. Here is the result for the Bigram model before and after normalization with Hazm:

Before normalization: ACC 85%



After normalization: ACC 75%

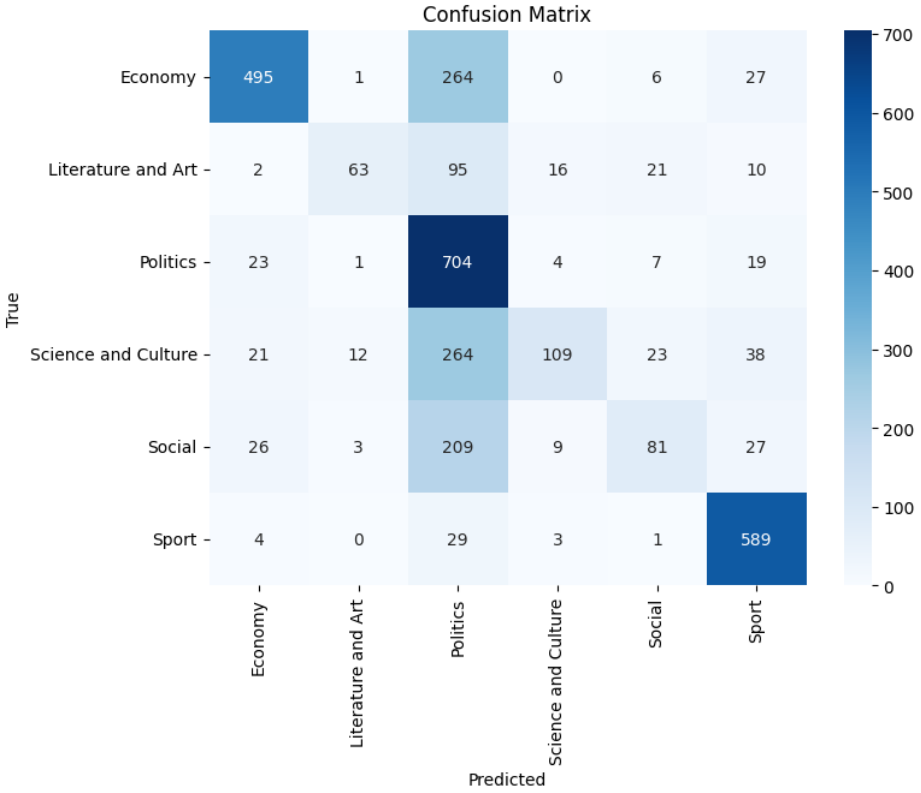


It may seem weird! But I decided to just skip the normalization step for these models, based on this observation.

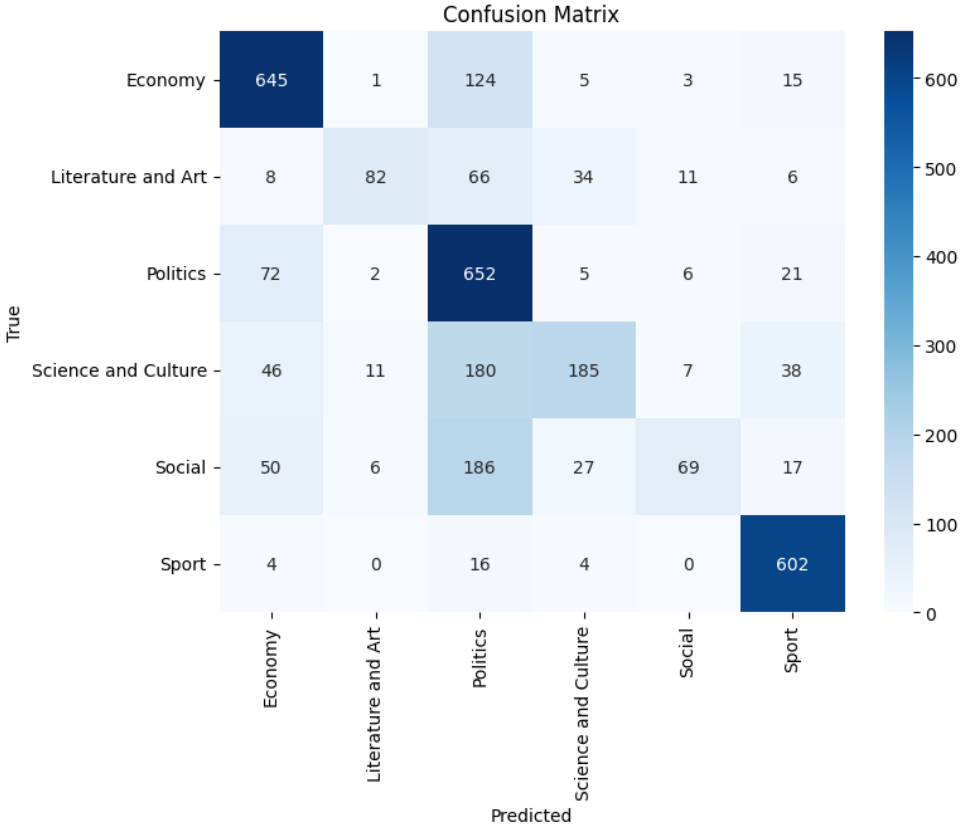
Stopwords

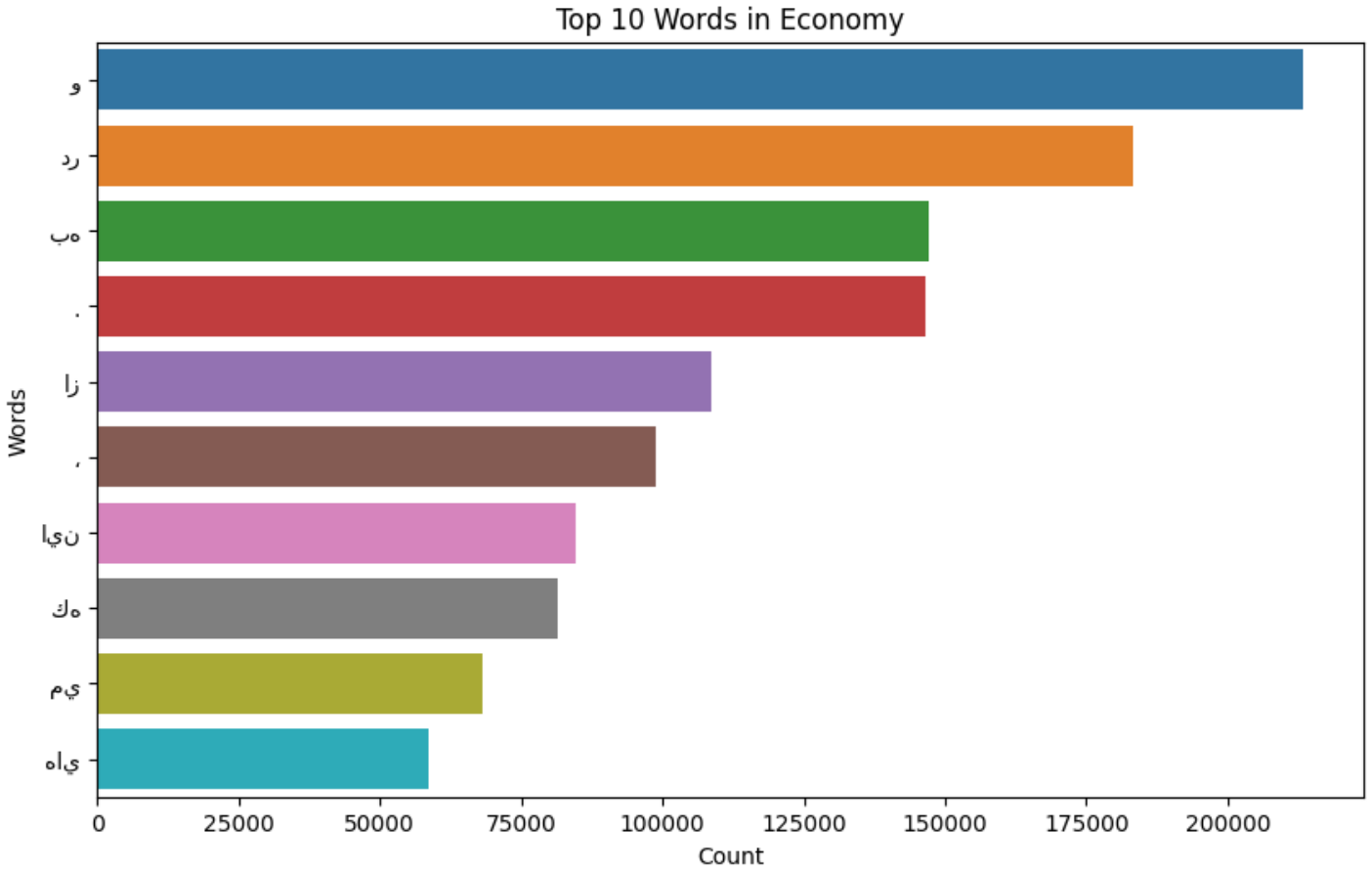
I have decided to try another pre-process in order to boost the accuracy of the model. This one could help me to accomplish that. Here is the result of the unigram model at the word level:

Before deleting stopwords: ACC 63.6%



After deleting stopwords: ACC 69.7%





I plotted the top 10 words in each category, and found out that there are many prepositions that are repeated many times! The solution was using a stopword list like the one Hazm prepared.

Unseen N-grams (Smoothing)

I have used a technique called “Laplace smoothing” to handle those n-grams that might not be in our train dataset. Without smoothing, if we encounter an n-gram in a test text that did not appear in the training data, the probability of that n-gram would be zero. This leads to an overall probability of zero for the entire sentence. So, smoothing can help to assign non-zero probabilities to unseen n-grams to prevent this issue.

Modern Models

BERT

We were asked to use the BERT model to perform a classification with the “Hamshahri News” dataset. I have used a famous library called “SentenceTransformers” and one of its pre-trained BERT models called “bert-base-multilingual-cased”. The reason to choose this model is that it is trained in 50 languages, such as Arabic.

Feature Extraction

At first, I used a BERT model to extract features from our dataset and then classify them with a Logistic Regression. Its accuracy was reasonable. For this type of classification, I used the whole dataset. You can see its results on the test set in the below images.

Only normalization:

	precision	recall	f1-score	support
Economy	0.91	0.93	0.92	3967
Literature and Art	0.82	0.82	0.82	1037
Politics	0.86	0.87	0.87	3788
Science and Culture	0.79	0.83	0.81	2336
Social	0.77	0.67	0.72	1776
Sport	0.98	0.98	0.98	3132
accuracy			0.88	16036
macro avg	0.86	0.85	0.85	16036
weighted avg	0.88	0.88	0.88	16036

After applying stemmer and stop words:

	precision	recall	f1-score	support
Economy	0.92	0.92	0.92	3967
Literature and Art	0.82	0.81	0.81	1037
Politics	0.86	0.88	0.87	3788
Science and Culture	0.79	0.82	0.80	2336
Social	0.75	0.67	0.71	1776
Sport	0.99	0.98	0.98	3132
accuracy			0.87	16036
macro avg	0.85	0.85	0.85	16036
weighted avg	0.87	0.87	0.87	16036

As it shows, It does not change too much!

Fine-tune

This is another way of using pre-trained models like BERT. At first, I used the whole dataset to train on, but each epoch took 21 minutes. Due to the limited time that I had, I decided to just use 20 percent of the training dataset. Here is the result of the fine-tuned model on the test dataset:

	precision	recall	f1-score	support
0	0.80	0.90	0.85	1037
1	0.99	0.98	0.98	3132
2	0.82	0.82	0.82	1776
3	0.94	0.96	0.95	3967
4	0.95	0.91	0.93	3788
5	0.87	0.85	0.86	2336
accuracy			0.92	16036
macro avg	0.90	0.90	0.90	16036
weighted avg	0.92	0.92	0.92	16036

As can be seen, the accuracy is much better than the feature excitation method.

Environment

Due to some conflicts between Hazm and other libraries in Google Colab, I decided to make an isolated virtual environment to avoid errors.

```
%mkdir neshan
! pip install virtualenv > /dev/null
! virtualenv neshan > /dev/null
! neshan/bin/pip install hazm > /dev/null
```

Thanks for your time and thoughtful consideration!