

Federal State Autonomous Educational Institution of Higher Education  
"Peter the Great St. Petersburg Polytechnic University Institute of  
Computer Science and Technology  
Mathematical support and administration of information systems

# Coursework

by discipline "Specialized Topics".

Summarization text, classification text and text-to-speech.

**Team:**

Mironov Vasiliy (Project manager),  
Vashilko Alexey (Backend),  
Vasilev Vadim (Backend),  
Tran Hai Long (Frontend),  
Ha Hoang Giang (ML Engineer)

**Group: 3530203/20002**

**Professor:**

Espinola Rivera Holger Elias

# 1 Problem Statement

Dataset "News Category Dataset" contains around 210k news headlines from 2012 to 2022 from HuffPost. This is one of the biggest news datasets and can serve as a benchmark for a variety of computational linguistic tasks. HuffPost stopped maintaining an extensive archive of news articles sometime after this dataset was first collected in 2018, so it is not possible to collect such a dataset in the present day. Due to changes in the website, there are about 200k headlines between 2012 and May 2018 and 10k headlines between May 2018 and 2022.

In this paper, three problems are reviewed:

1. Summarization text news
2. Classification shorted text news
3. Text-to-speech (shorted information about news in audio)

## 1.1 Summarization

The summarization algorithm translates the news text from the link specified in the dataset into a short text description that contains the main meaning of the news.

## 1.2 Classification

The classification algorithm takes a short description of the news and classifies the news according to 42 labels (categories). Thus, the problem of N-class text classification is solved.

## 1.3 Text-to-Speech

The text-to-speech algorithm takes the concatenated title, author, and short description news and translates it into an audio file.

# 2 Methodology of Solution

Each problem has its own approach. For the summation task, the ready-made **google-t5/t5-small** algorithm is further trained. For the classification task, two classifiers are implemented based on a fully connected neural network and LSTM. For the task of converting text to an audio file, a ready-made neural network from hugging face is used.

## 2.1 Infrastructure

To create the models, the following were used:

- Google Colab for training models on GPU's provided by Google.
- GitHub - repository with code.

- Kaggle provide dataset for training.

## 2.2 Summarization

The summation is performed on texts taken from links in the dataset using the **beautifulsoup** parser. Full texts of news items and their short descriptions are tokenized and then used by **Seq2SeqTrainingArguments**, **Seq2SeqTrainer** to further train the model. The summator is trained based on the ready-made **google-t5/small-t5** summator.

## 2.3 Classification

For the classification task, the **sentence\_transformer** library is used to compute the embedding of a short text description. The embedding is then fed to the input of a fully connected neural network or LSTM (Long Short-Term Memory). After which the classification into 42 categories occurs. Classification algorithms are implemented on **pytorch** libraries.

## 2.4 Text-to-Speech

A ready-made chugging face model called **suno/bark-small** will be used to translate text into speech. The model is used using the pipeline function from the transformer library.

# 3 Results

## 3.1 Summarization

During the summator training, an unknown error occurs, which interrupts the operation of the virtual environment kernel, and the google colab environment runs out of RAM. Therefore, similar to the text-to-speech task, a ready-made model was used

## 3.2 Classification

### 3.2.1 Deep Neural Network

Criterion: CrossEntropyLoss, Optimizer: Adam, Activation: Sigmoid.

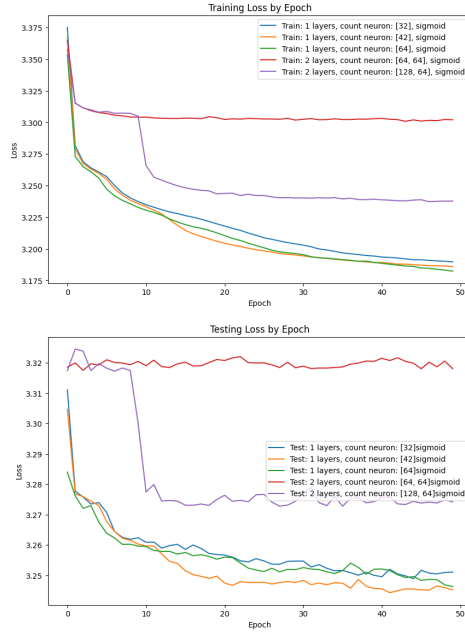


Figure 1: Training and test loss function dnn

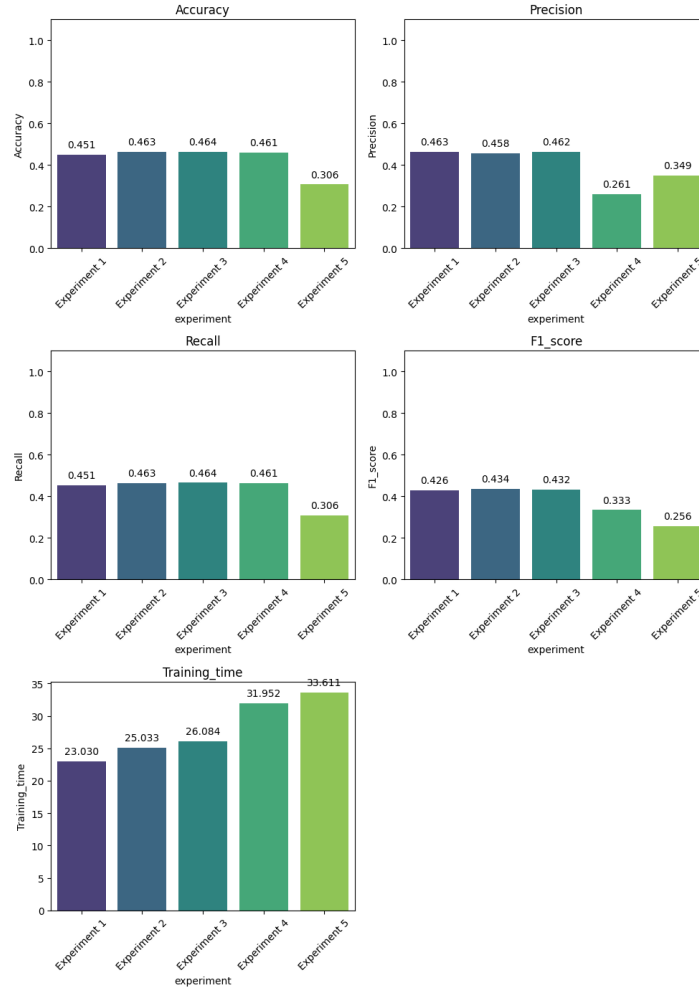


Figure 2: Accuracy, precision, Recall, F1 and training time comparison.

The best model with one layers and 64 neuron shows accuracy - 0.4640, precision - 0.4622, recall - 0.4640, F1 Score - 0.4324.

### 3.2.2 LSTM

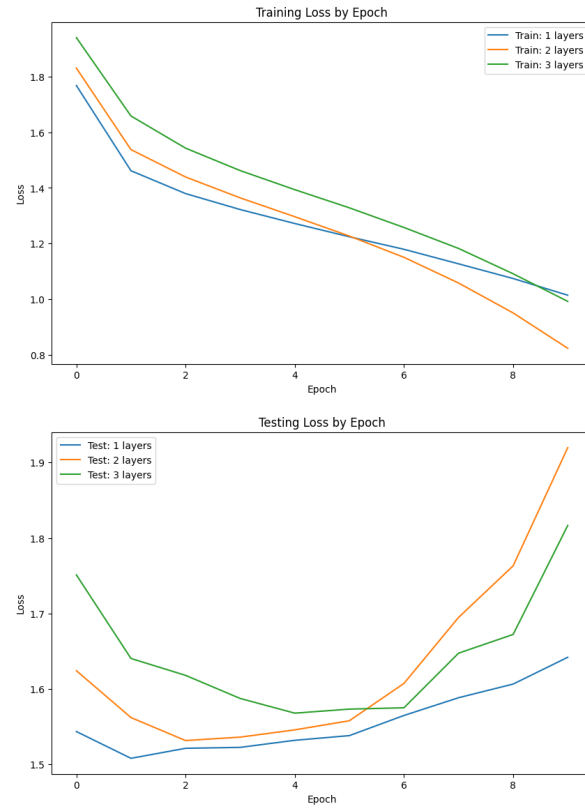


Figure 3: Training and test loss function lstm

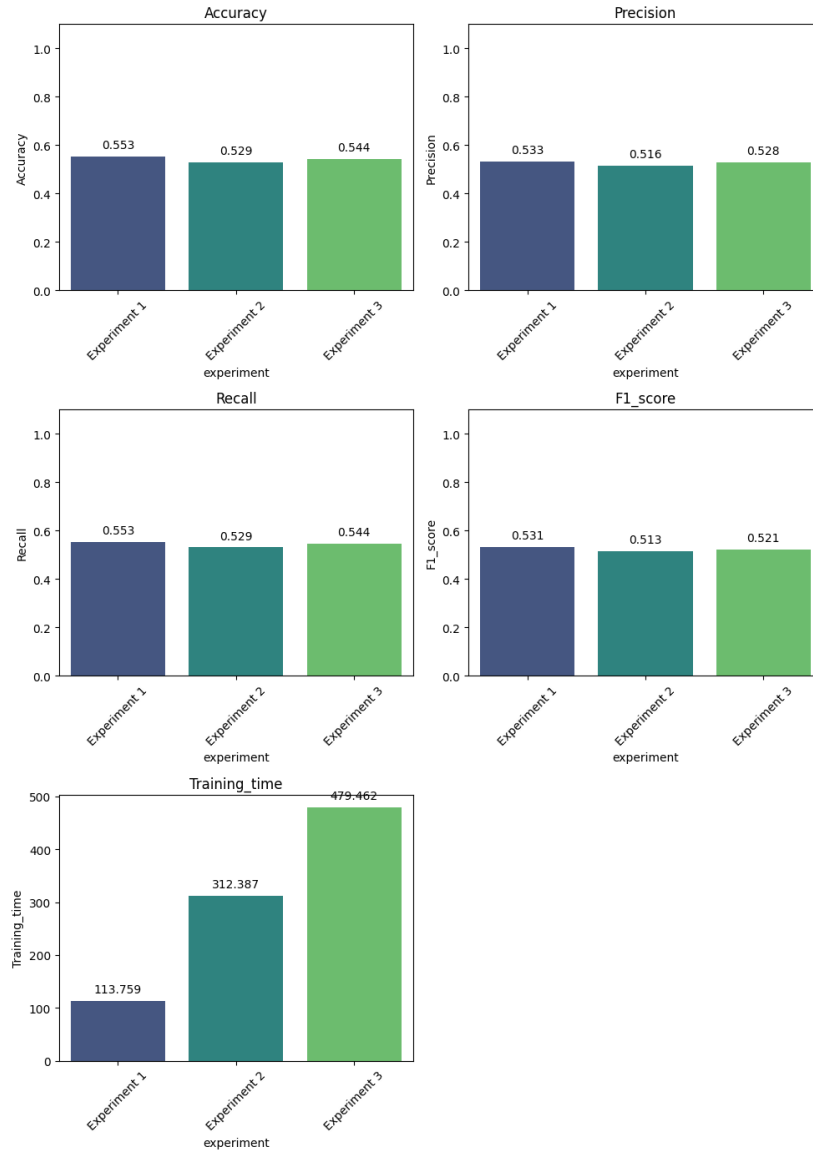


Figure 4: Accuracy, precision, Recall, F1 and training time comparison.

The best model with 2 hidden layers shows accuracy - 0.5535, precision - 0.5328, recall - 0.5535, F1 Score - 0.5313

As we can see, the LSTM model performs classification better than the fully connected neural network, because of tags and eventually it's starts to retrain.

### 3.3 Text-to-Speech

As a result of the algorithm's work, an audio file with a short audio recording of the news is obtained.

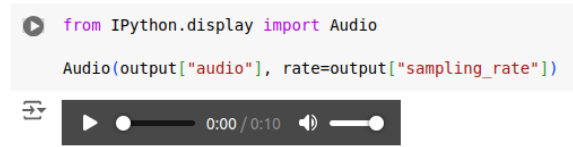


Figure 5: Result audio file of working text-to-speech

## 4 Conclusions

As a result of the course work, two models of classification of short text news were made and two models for summarization and text-to-speech conversion were applied. The following libraries were used in the work: **pytorch**, **transformers**, **sentiment\_embeddings**, **pandas**, **numpy**, **sklearn**, **requests**, **bs4**.