

Transformers and Attention-Based Deep Networks

Assignment - 1

Graduate School of Informatics

14.03.2025

Introduction

The objective of this assignment is to increase familiarity with transformer architecture and various adaptations for different applications. Your task is to analyze the sentiment of customer service conversations. You will begin with the dataset and apply explanatory data analysis, extract useful information about the data, and apply pre-processing steps necessary for the task.

You can find a starter code that is a modified nanoGPT implementation [here](#). The purpose of the starter code is to get you started on the transformer implementation. After understanding the codebase and the associated starter code you will be implementing your first working example. You will apply this first model for sentiment analysis after training. For the second model, you will obtain the pre-trained model weights of gpt-2 and fine-tune it. Note that changing the model structure too much will render you unable to fine-tune over a pre-trained gpt-2 weights.

Input: A customer-support interaction text. **Output:** 3 Classes for positive, negative or neutral

Deliverables

Your submission is expected to be uploaded to ODTUCLASS in a single zip file with the naming convention “DI725_assignment.1_STUDENTNUMBER.zip”.

Report: Prepare a report that clearly explains the steps in this assignment, changes and tweaks over the starter code and how these affected the results. The report shall have the following sections: abstract, introduction, dataset, modeling, evaluation, results, discussion, conclusion.

Code: The code that is to be shared must be working and reproducible, submit your code to odtuclass, and also upload it to your public GitHub repo.

Version Control: Your implementation shall be traceable from your [github](#) repo. Your repo shall be public so that we can examine your code.

Experiment Tracking: Experiments in this assignment are to be recorded to [WANDB](#). Your WANDB experiments page or WANDB report page shall be public so that we can examine your experiments.

Format

The report document must be a pdf file. The report shall be in IEEE format. You can find the IEEE template [here](#). Please follow this path to select the correct format: Conferences, Original Research, Word or Latex. Software can be shared as a project consisting of python scripts (.py) or an interactive notebook (.ipynb). If you are using any online interactive notebook service (Google-Colab for example) for your coding part, you can download it as a notebook that will preserve your code and comments and markdown text.

Duration

This is a two-week assignment that should not be rushed in the final days. Any submission after the deadline will have a penalty of 10 points for each day. You will be unable to submit after 2 days.

Assignment

Let's assume we are working in the automation department of a company. We are tasked with evaluating our customer service! We will be analyzing the sentiment of our customers based on their brief interaction with our customer service.

Dataset

The dataset we are going to use is available [here](#). There are two files for the dataset, one that is labeled as train is to be used for training and validation purposes. The one labeled test is to be used only for testing and should not be leaked into the training.

Sentiment Analysis

In this part, it is your task to use the dataset and starter-code according to the following instructions:

1. Set up Weights and Biases (WANDB) for your own use, create an account if you don't have one yet. WANDB is an experiment tracking tool that can also be used to automated reporting and hyperparameter tuning.
2. Set up GitHub account for version control. Git is a mandatory skill for any software developer and crucial for version tracking.
3. Modify the shared code for WANDB, i.e., input your specific WANDB API key (first time only if using a local machine). The code has the necessary imports and arguments for WANDB use.
4. Download the dataset. Note that there are train.csv and test.csv. Do not leak any data from test.csv, it is a very small portion of the original dataset. The dataset is a customer service interaction data where customers define their problems and agents of the company try to help them.
5. Prepare an Exploratory Data Analysis for the dataset, indicating the distribution of the sentiment classes and other important facts about the data. Is there any correlation between other features and the sentiment feature? Note that this part is a very important step that helps with understanding the data we are using and getting useful insight into the problem.
6. We are going to use customer_sentiment and conversation features, omit other columns. And prepare your train and validation subsets. Do not leak data from the test set.
7. Data pre-processing: Explain which pre-processing steps you have taken for the dataset and justify your approaches. Hint: Are you thinking about using all of the text in the conversation field?
8. Redefine and modify the transformer architecture (nanoGPT) to fit sentiment analysis task.
9. Train a model from scratch following the nanoGPT implementation and your modifications for the sentiment analysis task.
10. Apply fine-tuning over the pre-trained gpt-2 model to the sentiment analysis task with your modifications and redefined architecture. You can check the starter code on how to fine tune over pre-trained model.
11. Report the results using at least two appropriate evaluation metrics for this task and confusion matrix. Give your reasons for your choice of evaluation metrics.

Report

Your report shall contain the following information:

Abstract: prepare an abstract (100 words max)

Introduction: Briefly introduce the task.

Dataset: Explain and present pre-processing steps and exploratory data analysis briefly.

Modeling: Explain the changes made to the model (nanoGPT) architecture. Discuss the modifications and their benefits for the task.

Evaluation: Discuss which evaluation metrics were used in the assignment and why did you choose them.

Results: Share the results of the experiments as a summary of WANDB experiment tracking tool.

Discussion: Compare fine-tuning models over a pre-trained model and models trained from scratch. Discuss your findings and results.

Conclusion: Conclusion and final remarks.

Grading

1. **Data exploration and pre-processing (20 pts):** You shall report the pre-processing steps you have taken in this assignment. You shall also prepare a concise and brief exploratory data analysis, figures, tables and plots to augment your report.
2. **Version control (Git) (10 pts):** your GitHub repository shall be public and contain at least 3 commits in different days. If you encounter an issue or a bug while doing this assignment you shall open an issue describing it. If you leave this assignment to the last day you will not get any points from version control.
3. **Experiment Tracking (WANDB) (10 pts):** your wandb experiments or report page shall be public. If it is not public you will not get any points from experiment tracking.
4. **Training from scratch (20 pts):** your code shall be a modification of the provided starter code. Apply common clean coding practices and comment where necessary to get full points. If your code is messy and hard to understand you will not get full points.
5. **Fine-tuning (20 pts):** Fine-tune over a pre-trained model. Apply common clean coding practices and comment where necessary to get full points. If your code is messy and hard to understand you will not get full points.
6. **Report (20 pts):** Your report shall be 2 pages maximum, if you exceed this limitation you will get -5 points for each extra page. Your report is your only way of communication for your assignment, it shall explain clearly all necessary steps you have taken. Your report shall have the following parts: abstract, introduction, dataset, modeling, evaluation, results, discussion, conclusion.