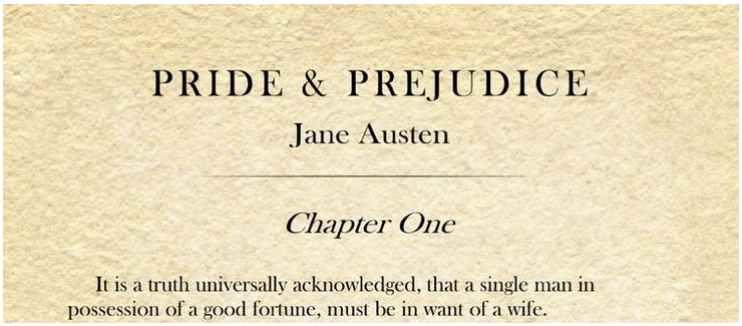
COSC420 Assignment 2

Transformer-based language model

Weight:20% Lecturer: Lech Szymanski

For this assignment, you will be building and training neural network models using Tensorflow's Keras library.

Dataset



The dataset for your training and testing is the text of "Pride and Prejudice" (over 100K words) and/or the text of "War and Peace" (almost 600K words).

Task 1: Vector representation of text tokens (6 marks)

Build a train a neural network for Word2Vec encoding of the text tokens generated by the provided BPE tokeniser... and since these tokens are not necessarily always entire words (akin to FastText), I'll refer to this encoder as Tok2Vec. It is up to you to decide the dimensionality of your vector embedding, whether to use the CBOW or skipgram approach, etc. You may use for training either the text of "Pride and Prejudice", or "War and Peace", or both.

Task 2: Text prediction with a transformer (8 marks)

For this task, you are to build and train a *language model* using transformer architecture on the text of "Pride and Prejudice", or "War and Peace", or both. The building blocks of the transformer architecture for Tensorflow will be provided, but it is still up to you to make various choices about the hyper-parameters (such as size of the input sequence, number of layers in the model, heads in self-attention, neurons in the dense networks, etc.) as well as the details of the training. Your model does not need to generate the `<|endoftext|>`

output; for evaluation choose a fixed number of tokens for the model to generate after some starting prompt. Train the model on two variants of text encoding:

- one-hot encoding
- your Tok2Vec encoding from Task 1

I am not expecting ChatGPT-level quality of the generated text – it simply can't be done without the massiveness of the Large Language Model and the training data that contains good chunk of the Internet. What I value is a system that can work in principle and does something that does some kind text of generation, even if it non-sensical (or comical).

Task 3, Report (6 marks)

Write a report of what you have done. What I am looking for here is a short description of methodology and results for each task. Methodology should be a short summary of the architecture used, hyper-parameters and decisions with respect to aspects of training. For the results I'd like to see:

- Task 1 – visualisation of the embedding through either t-SNE dimension reduction (for which the code will be provided) or K-means clustering;
- Task 2 – the training (and validation if you'd like to include it) accuracies of the models with two encoding variants, qualitative assessment and comparison of the the text generating capabilities of the two model variants. For instance, in the qualitative assessment you could generate some number of words after prompting the model with a phrase directly taken from the training text (such, the first part of the sentence, "It is a truth universally acknowledged" of "Pride and Prejudice", if trained on that text) and judge how well the generated text (for some number of words) mirrors the book. Then you could prompt with text that was not directly taken from the training, to see how well the model generalised to generating text in different scenarios. Compare the two model variants.

You should also comment about your results for each task.

Submission

The assignment is due at **11:59pm on Monday of 20th May**. For each task I expect submission of:

- Task 1: python code for training your Tok2Vec model as well as saved files with saved encoding (including the json file with saved state of the BPE tokeniser you used/built) – ideally, there should be a flag, which allows building and training the model from scratch or loading a pre-saved one (that needs to be included in your submission). Scripts for loading text data into Python and an implementation of BPE tokeniser are provided (see Blackboard, Assignments, `load_text.py`, `tokeniser.py`).
- Task 2: python code for training transformer models (could be in same file or two separate files) as well as saved files with the trained models weights; each script, again, should have a flag (like in the tfintro examples), which allows building and training the model from scratch or loading pre-saved (submitted) one. Script providing custom-built Tensorflow layers for embeddings, positional encodings and transformer layers with multihead self-attention is provided (see Blackboard, Assignments, `transformer.py`
- Task 3: a pdf file with your report; roughly around 1500 words; more words (if needed) are fine, but this is meant to be a technical report – concise, but clear; short descriptions of methodology and results analysis will suffice.

Zip the folder including your code, saved model files and report and submit electronically via Blackboard. Don't forget to clean up your code before submission and to add comments. Make sure to save your models after training and to include them in the submission.

Academic Integrity and Academic Misconduct

Academic integrity means being honest in your studying and assessments. It is the basis for ethical decision-making and behaviour in an academic context. Academic integrity is informed by the values of honesty, trust, responsibility, fairness, respect and courage. Students are expected to be aware of, and act in accordance with, the University's Academic Integrity Policy.

Academic Misconduct, such as plagiarism or cheating, is a breach of Academic Integrity and is taken very seriously by the University. Types of misconduct include plagiarism, copying, unauthorised collaboration, taking unauthorised material into a test or exam, impersonation, and assisting someone else's misconduct. A more extensive list of the types of academic misconduct and associated processes and penalties is available in the University's Student Academic Misconduct Procedures.

It is your responsibility to be aware of and use acceptable academic practices when completing your assessments. To access the information in the Academic Integrity Policy and learn more, please visit the University's Academic Integrity website or ask at the Student Learning Centre or Library. If you have any questions, ask your lecturer.

- Academic Integrity Policy
- Student Academic Misconduct Procedures

Use of ChatGPT as a tool to help with coding and the writing of the report is allowed, but only as an aid to, and not a replacement of, your effort. For instance, it is ok to accept Copilot's suggestions of your coding when trying to implement a model in Tensorflow/Keras of your design. It's not ok to ask ChatGPT to generate code that will create a complete Transformer network, especially if it is not using the components provided, for the described task.