**Program 2 – Report**

**Sam Timmins, Alex Cerpa, Kas Taghavi**

**Overview**

For our model, we used the transformer architecture and finetuned a pretrained model named distilBERT. We chose to use a transformer as we thought that the self-attention mechanism that allows it to focus on different parts of the input sequence would be well-suited for our problem. The large contextual embeddings contained within the pretrained models also provide relationships between words that can be useful for determining word-sense. We created one model per word, framed as a classification problem where each sense was its own class.

**Preprocessing & Training**

We used regular expressions to handle parsing the original input files given to us into dataframes. We removed all special symbols that were not square brackets (as they are used within the model for special tokens). Trailing and leading spaces were also removed. We utilized a 90/10 training and test split on each word. We sourced additional sentences from online resources such as the discussion board for our class and print media. For a rudimentary validation set, we utilized the LLM, ChatGPT, to generate sentences for us, given each word and word sense combination.

The input of the model was formatted as the following:

<input sentence> [SEP] <word> [SEP] <sense\_1> [SEP] <sense\_2>

Here is a simple example for the word “yarn”:

The yarn is no longer novel too many other writers have since taken off from Galluns inspiration but it is just as fine to me as it always was [SEP] Yarn [SEP] 1 n narration recital yarn the act of giving an account describing incidents or a course of events his narration was hesitant [SEP] 2 n thread yarn a fine cord of twisted fibers of cotton or silk or wool or nylon etc used in sewing and weaving

We used the preexisting special tokens within the pretrained model to be able to utilize it for classification. The [SEP] token denotes when one section of text ends and another begins. We trained the model for 3 epochs with a batch size of 8.

We used the transformers library to handle model loading, model training, and model saving. The transformers library already provides preexisting models for sequence classification, which is what we used. For inference, we used the pipeline module within transformers.

Text, letter

Description automatically generated