TF-IDF firstly creates a vocabulary of words. It then creates a term frequency vector per document containing the frequency of each word from the vocabulary in the document. These are normalised by the inverse document frequency that assigns greater weighting to rare words under the assumption they convey more unique information about a document.

An advantage of TF-IDF is that is easy to compute, and we can use cosine similarity to compare documents easily. However, it is a bag of words model, so we lose information about the order of the tokens which is valuable. These vectors are also high-dimensional and sparse which is expensive for computation.

I initially use BERT for feature extraction. Instead of a vocabulary of words, BERT uses subword segmentation called WordPiece. This breaks words down into pieces based on their value to the language model. Instead of training BERT or WordPiece, I use the HuggingFace’s implementation to extract input IDs. I strip excess whitespace, URLs, HTML, and emojis and then apply the tokenizer to the headlines and bodies individually. This extracts token IDs and attention masks (for short sequences).

Using these, I concatenate the headline and bodies by shortening the bodies to fit within the limit and joining them with a [SEP] token. Now, I use the pre-trained BERT to extract the [CLS] token which summarises the information from the model and can be used to train the later models.