CNN

Urmzd Mukhammadnaim

December 5, 2021

Contents

Chapter 1

Methodology

1.1 Data Processing

After the extraction of the documents for each classification, we explored two preprocessor implementations. We first describe the use of One Hot Encoder and Witten-Bell Probability Distribution to retain morphological-level information, and generate unique sets of words that can later be fed into the CNN. Subsequently, we analyze the use of the FastText model as a means of retaining semantic-level information, and ensuring words with similar words appear closer together in the subspace.

1.1.1 One Hot Encoding and Witten Bell Generation

After the documents were tokenized, the text was piped into the transformer $T1$ which applied a series of filters and maps to remove punctuation and ultimately break the words into their stems. The result of $T1$ would consist of a set of unique stems which could be fed into the encoder D_n where n represents the order of which the the document was processed in. After $T1$ was applied to all documents, the sets were unioned to generate the vocabulary V. V was then fed into a *FreqDist* class from the *nltk.probability* package to allow the subsequent usage of the *WittenBellProbDist* class.

Using the OneHotEncoder module from sklearn's preprocessor package,