Introduction to Gnearative AI and preprocessing : Assignment

Score: 58 / 61

✓

The text preprocessing objective is to process the raw data and make it suitable for use in the Gen AI model. It involves processes like cleaning (removing stop words, white spaces, unwanted characters, punctuations, tags, making lower case), lemmatization, stemming, tokenization, pos, and ner tagging.

Feedback

Great answer! You have provided a comprehensive explanation of the primary objectives of text preprocessing in the context of generative AI. Well done!

✓

Tokenization is the process of breaking down a sentence or text into smaller units called tokens. These small units can be at the word level, sub-word level, or character level. Each gen AI has a specific method of tokenization. These tokens are then embedded as dense vectors for processing through the Gen AI model. We must use the same tokenization that was used during training when we use the LLM model for either fine-tuning or inference.

Feedback

Great answer! Well explained and demonstrates a deep understanding of tokenization in the context of generative AI.

The text cannot be fed in the Gen AI model as it is. It needs to be converted to numerical format before being fed. One hot encoding is one of the many techniques to represent text in numerical form .

In one hot encoding, we first tokenized the sentence/text. We build a vocabulary of unique tokens. We then use binary labels for each token where all labels are zero except for the one token that we need to represent. So let's say we have a vocabulary of 10000 words. We will use a vector of 10000 dimensions to represent each token. The vector will have a value 1 only at the position which indicates the token and 0 elsewhere. The vector will be sparse.

Feedback

Great explanation of one-hot encoding and its relevance to text preprocessing in generative AI models. Well done!

✓

Word embeddings help in the dense representation of words along with other semantic meanings. Unlike one hot encoding which represents token using sparse vectors word embedding represents token using continuous vectors along takin semantic meaning into account. Popular word embedding algorithm include Word2Vec and Glove

Feedback

Great answer! The explanation is clear and demonstrates a good understanding of the role of word embeddings in text preprocessing for generative AI. Well done!

✓

Data cleaning helps to deal with noisy and raw data. It converts them to a format that can be easily processed by the model. They also help to ensure that any biase or incomplete or wrong information is not passed to the model which is crucial for fairness.

Normalization techniques help to standard the input text data. This increases the quality and consistency of the training data which leads to better output from mode.

Feedback

The answer provides some understanding of data cleaning and normalization, but it lacks depth and specific examples. It needs to explain the importance of these processes in the context of text preprocessing for generative AI in more detail.

✓

N-grams are contiguous sequences of n items (words or characters) extracted from a given text. They help in capturing local context and prepare feature-rich vectors for the numerical representation of text compared to one hot encoding. We can use a higher order of n for capturing longer dependences. They help in the probabilistic modeling of a sequence of words in language modelling

Feedback

Great answer! You have demonstrated a clear understanding of how n-grams contribute to improving the quality of generated text in generative AI models. Well done!

✓

Noisy data poses the following challenges

Unambiguity - Their meaning cannot be deciphered with confidence. Noise data could contain misspellings, abbreviations, incomplete text, slang etc. The tokenization and embedding technique will either handle it incorrectly or replace it with the unknown token.

Unexpected output - IF not handled properly during pre-processing noisy data reduces model effectiveness in producing correct output

Potential solution include

Spell checker

Train model to take into account common accepted abbreviation

Context-driven imputation for incomplete text

Feedback

Great answer! The response effectively outlines the challenges of handling noisy or unstructured text data for generative AI models and provides potential solutions. Well done!