CSCI 5832 Homework 1 Part 2

Tuguluke Abulitibu

August 31, 2021

How many words does BERT know?

I consider this task as a continuation of part 1. The answer is highly corelated to the definition of 'word'[1]:

1. a single distinct meaningful element of speech or writing, used with others (or sometimes alone) to form a sentence and typically shown with a space on either side when written or printed.
2. a command, password, or signal.

In part one, I did analysis on only one sample of text data (without considering any 'command, password, or signal') and gave an estimate of how many I know. In part 2, I will do more or less of a conversation with myself so to write down what I think I've leared about the denition of 'word', so far.

Approach 1: Do nothing

The text file loading shows it has 30522 entry, we can 'argue' every character, every sign(signal), along with their 'location', is unique, That gives us the original **30522**. If we run a unique function, that number will drop down to **29498**, when we don't consider the repetition of words.

Approach 2: No 'emoji's

Athough we are in the age of 'emoji's, I still don't consider special characters as 'words'. Justing by simply 'stripping off' these formula, we can get the number down to **25975**.

Approach 3: Lemmatization

Finally, I've decided to just focus on the meaning of 'a single distinct meaningful element of speech or writing', the dictinctness, in my current openion, should dictates how many words BERT know. A good example (or bad) is the word 'investigate'. Apparantly the text has 10 words corelate to this one, such as: investigate, investigated, invesigating, etc. But the uniqueness should be in the 'word stem' itself, otherwise it would be any number that we imagine, hence we can find the diffrent tenses of each word and claim we simply multiple by that number with the number of unique words. Curently, I believe that Lemmatization should be as the threshhold of diffrent works. With that in mind, the number will count down to **21342**. I've also noticed that some words has no meaning unless they are shorten for something, works like 'sa' (sweden?), ti (Italian?). After getting rid of those the number drops down to **20974**.

I am sure there are no 'correct answer' for this assignment since there are many more methods to counts (to refine), I am looking forward to learn more in this class.

[1] google.com

What's missing?

When I was learning Russia, I thought I can just grasp the most of it by memorizing the words, what a disaster that was! I know nothing about a word until I can locate the work in a real sentence. Same rule should be applied here, BERT does not any words until it considered them through sentences. And that is the missing piece in this assignment, we need the 'know' the word contexually, that is, within a real sentence or a paragraph. So a better way for BERT to know each word should be from a collection of texts, not just line of words (although in this case we get rid of word like 'to', and 'am'. but there are not that many there in the first place).

In conclusion: Purely numerically speaking

Approach	Number of words
Do nothing	30522 (29498 unique)
No 'emoji's	25975
Lemmatization	21342 (20974 if getting rid of 2 letter words)

Code snippet

```python
# Transform text to lower case, remove unnecessary punctuation if present
def regex_clean(text):
  cleaned_text = re.sub(r"[^a-zA-z]", " ", text.lower())
  return cleaned_text

def tokenize(text):

  #Ensure type is string
  text = str(text)

  #Make lower-case, remove punctuation and extra spaces, tokenize into words/phrases.

  #Use 'split' instead of 'word_tokenize' to have space
  tokens = str.split(text, '::')
  tokens = [regex_clean(tok) for tok in tokens]

  #Remove stopwords,
  tokens_no_stops = [tok.strip() for tok in tokens if tok.strip() not in all_stopwords]

  #Remove words with only 1 letter
  tokens_large = [tok for tok in tokens_no_stops if len(tok)>1]

  #Initialize Word Lemmatizer
  lemmatizer = WordNetLemmatizer()

  #Lemmatize
  tokens_lemmatized = [lemmatizer.lemmatize(tok) for tok in tokens_large]

  return tokens_lemmatized
```