Final NLP class Project Report

# BBC News Analysis

Mahsa Ghaderan

Supervise by

Dr Sauleh Etemadi

June 23, 2021

**Department of Computer Engineering**

**Iran University of Science and Technology**

# Contents

# List of Figures

# List of Tables

# Part 1

# Word2Vec

In this part I used assignment2 from cs224n, 2021 Stanford NLP course as my base code. Different Word2Vec models are trained for each label which exist in BBC data set. Headline is concatenated with body of each news for as input text for the model. Each models trains for 40000 iteration. It took about 5 hours for each label. Wights of models are saved is ”models/word2vec” directory. Most 30 repeated words are chosen from each label. Images - , - , - and - shows distribution of each model on a 2D map.

## 1.1 Iran News

## 1.2 Art News

## 1.3 Sport News

## 1.4 Economic News

## 1.5 Science News

# Part 2

# Tokeniation

SentencePiece is an unsupervised text tokenizer and detokenizer python project, which mainly used for text generation tass.This part ia mainly based on this packet. It use a model to train what tokens are best to tokenizing a given corpus. In this part 2 different methods of tokenization in implemented. First is tokenizing words and sub-words, in this methods model looking for most repeated words and sub-words and even letter. As a result we can see almost every word in final version of chosen tokens. On the contrary the second method is just tokenizing words, this models looking for most repeated words.

Until this part entire headline and body of each news are both covered in the corpus but from this part English values are removed from corpus so tokenization and language generation will focus on Persian.

The corpus is shuffled 5 times and separated to train and test data. Also tokenization is applied on 4 different vocab size due to possible SentencePiece size and performance of output model is evaluated against these vocab size. In the following section, we discuss about effects of vocab size from very small value increasing to a large value.

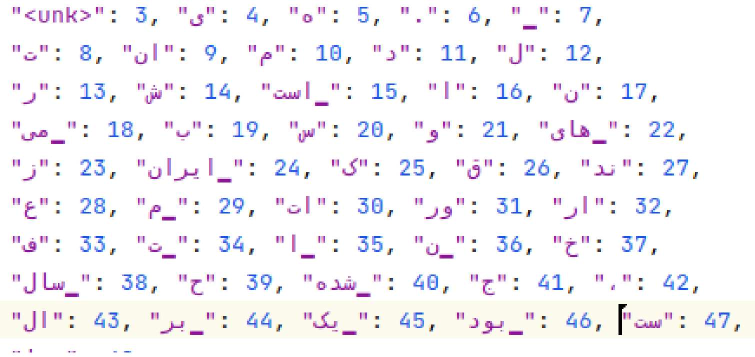
## 2.1 Sub-word Level

At first i train tokenizer model on sub-words. It means model not only pays attention to words, but also pays attention to sub-words during training time.

When the vocab size is 1000, it is too small for such a corpus and due to model type, model focuce on smaller sub-words rather than meaning full words. In final choosed-vocab, almost every character in source language exists. So when we tokenize a text

there woudln't be any <unk> token and most words break into sub-words.
As vocab size increases model learns more complicated sub-words and less meaning-less characters and sub-words. Bot the negative point is as the vocab size increases it would be harder for models to learn task and the positive point is more context will be save after tokenization.

Another point is SentencePiece can discriminate between statrting token and not starting token and ending token of a word in this mode. As we can see in 2.1 the are different form of one token.



```
"<unk>": 3, "ی": 4, "ه": 5, ".": 6, "_": 7,
"ت": 8, "ان": 9, "م": 10, "د": 11, "ل": 12,
"ر": 13, "ش": 14, "_است": 15, "ا": 16, "ن": 17,
"_می": 18, "ب": 19, "س": 20, "و": 21, "_های": 22,
"ز": 23, "_ایران": 24, "ک": 25, "ق": 26, "ند": 27,
"ع": 28, "_م": 29, "ات": 30, "ور": 31, "ار": 32,
"ف": 33, "_ت": 34, "_ا": 35, "_ن": 36, "خ": 37,
"_سال": 38, "ح": 39, "_شده": 40, "ج": 41, "،": 42,
"ال": 43, "_بر": 44, "_یک": 45, "_بود": 46, "ست": 47,
```

Figure 2.1: Sub-words vocab example

## 2.2 Word Level

In this part, i used word level tokenization, in this experience there exist <unk> token due to vocab size and model dealing with choosing best token in order to minimize <unk> token in unseen text.
Same as previous experiment, data-set is divided into train and test set and 5 different shuffled corpus is saved, this help us to have a better evaluation on each vocab size. Four different vocab size is defined as follow: 1000, 5000, 10000, 15000. Number of <unk> token and percent of that over each corpus and vocab is reported in ”report/tokenization” directory and each model and output on test set is saved in ”src/tokenization/working-dir/words-out”. percent of each vocab size is presented in table 2.2.

| | word-level | | | | | |
|---|---|---|---|---|---|---|
| corpus | 1 | 2 | 3 | 4 | 5 | AVG |
| 1000 | 0.44 | 0.44 | 0.44 | 0.43 | 0.44 | 0.44 |
| 5000 | 0.20 | 0.20 | 0.20 | 0.20 | 0.20 | 0.21 |
| 10000 | 0.14 | 0.14 | 0.14 | 0.14 | 0.15 | 0.15 |
| 15000 | 0.12 | 0.12 | 0.12 | 0.12 | 0.14 | 0.13 |

As it is illustrated in table 2.2, number of <unk> token increase by vocab size reduction. the fewer <unk> token, the fewer information loss. On the other the larger vocab size leads into more amount of computaion for future task. As we saw it is a trade of between these to parameters, in my opinion best vocab size due to this corpus is **10000**. It has reasonable <unk> token and it may need less computation rather than 15000 vocab size.

# Bibliography

[1] D. Evans, P. Gruba, and J. Zobel, *How to write a better thesis.* Melbourne Univ. Publishing, 2011.

[2] H. Kopka and P. Daly, “A guide to {\LaTeX}–document,” *Journal name*, 1995.

# Appendices

# Appendix A

# Appendix Title

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.

This is the second paragraph. Hello, here is some text without a meaning. This text should show what a printed text will look like at this place. If you read this text, you will get no information. Really? Is there no information? Is there a difference between this text and some nonsense like “Huardest gefburn”? Kjift – not at all! A blind text like this gives you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in of the original language. There is no need for special content, but the length of words should match the language.