In the name of God

Department of Computer Engineering

Natural Language Processing

Final Phase Report *

Yeganeh Morshedzadeh

Student Number: 96521488

Spring 2021

*https://github.com/yegmor/NLPProject

Contents

List of Figures

List of Tables

Abstract

In this project, we tried to use Natural Language Processing to better understand Depression and Anxiety posts. The dataset is gathered from Reddit communities r/depression and r/Anxiety.

For this project, at first, we wrote a project proposal (Google Docs), and afterwards, in the first phase (Google Docs), we gathered data and made some exploratory data analysis.

In the final phase, we went deeper, and tried various NLP tasks, such as, computing Word2Vec, Tokenization, Parsing, and creating a language model based on our the dataset.

Part I
Word2Vec

Filename: 3_word2vec.ipynb

Code

For this part we have three Word2Vec models, named as dep_w2v_model, anx_w2v_model, and all_w2v_model. Moreover, with boolean parameters, load and save, the model will be saved and/or loaded in the my_word2vec function.

Table 1: Word2Vec vocabulary size

	label	vocab_size
0	depression	2054
1	anxiety	2175
2	all	3223

Results and Examples

To make the visualizations more relevant, we will look at the relationships between a query word (in **red**), its most similar words in the model (in **blue**), and other words from the vocabulary (in **green**)

Part II
Tokenization

Filename: 4_tokenization.ipynb

Code

In this part we have used KFold to split our data into train and test. Afterwards, we train SentencePiece model based on the data. Lastly, we compute <UNK> on our test dataset.

Results and Examples

Part III
Parsing

In this part, we used Stanza, which is a a Python NLP Package, and a collection of accurate and efficient tools for the linguistic analysis of many human languages. Starting from raw text to syntactic analysis and entity recognition, Stanza brings state-of-the-art NLP models to languages of your choosing.

More specifically, we used their Online Demo to create a manual .CoNLL file based on our dataset. Later, we can use Universal Dependencies CoNLL viewer to automatically generate parse tree.

The depen

Part IV
Language Model

Filename: 5_language-model.ipynb

Code

In this part we have used KFold to split our data into train and test. Afterwards, we train SentencePiece model based on the data. Lastly, we compute <UNK> on our test dataset.

Results and Examples

Part V
Fine Tuning

Classification

Filename: 6_finetune_classification.ipynb

Code

In this part we have used KFold to split our data into train and test. Afterwards, we train SentencePiece model based on the data. Lastly, we compute <UNK> on our test dataset.

Results and Examples

Language Model

Filename: 7_finetune_language-model.ipynb

Code

In this part we have used KFold to split our data into train and test. Afterwards, we train SentencePiece model based on the data. Lastly, we compute <UNK> on our test dataset.

Results and Examples

References

[1] https://towardsdatascience.com/goodbye-world-4cc844197d51

[2] https://colab.research.google.com/github/google/sentencepiece/blob/master/python/sentencepiece_python_module_example.ipynb#scrollTo=ee9W6wGnVteW

[3] https://gmihaila.github.io/tutorial_notebooks/gpt2_finetune_classification/

[4] https://colab.research.google.com/github/philschmid/fine-tune-GPT-2/blob/master/Fine_tune_a_non_English_GPT_2_Model_with_Huggingface.ipynb#scrollTo=hKBSyNLgqF9K

[5] https://github.com/huggingface/notebooks/blob/master/examples/language_modeling.ipynb

[6] https://www.kaggle.com/pierremegret/gensim-word2vec-tutorial

[7] https://machinelearningmastery.com/how-to-develop-a-word-level-neural-language-model-in-keras/

[8] https://huggingface.co/transformers/training.html

[9] https://www.kaggle.com/achintyatripathi/gensim-word2vec-usage-with-t-sne-plot

[10] https://colab.research.google.com/github/borisdayma/huggingtweets/blob/master/huggingtweets-demo.ipynb

[11] https://huggingface.co/transformers/custom_datasets.html

[12]