Project 3:

Natural Language Inferencing

By Arunkumar Ramachandran(903928488),

Sagar Jagtap(903935334)

Shiv Vyas(903939075)

Instructor : Dr. Ryan White

MTH 5320

10/9/2020

1. Introduction

Natural language processing (NLP) has grown increasingly elaborate over the past few years. Machine learning models tackle question answering, text extraction, sentence generation, and many other complex tasks. But, can machines determine the relationships between sentences, or is that still left to humans? If NLP can be applied between sentences, this could have profound implications for fact-checking, identifying fake news, analyzing text, and much more.

If you have two sentences, there are three ways they could be related: one could entail the other, one could contradict the other, or they could be unrelated. Natural Language Inferencing (NLI) is a popular NLP problem that involves determining how pairs of sentences (consisting of a premise and a hypothesis) are related.

The main task is to create an NLI model that assigns labels of 0, 1, or 2 (corresponding to entailment, neutral, and contradiction) to pairs of premises and hypotheses.

2. Goal

The main goal of this project is to determine the relationship between different sentences by classifying the relation between the sentences according to their respective labels and train the model to obtain good accuracy. The assigned labels are 0, 1, or 2 (corresponding to entailment, neutral, and contradiction) respectively.

3. Dataset

There are two datasets used in this project. The links have been included for reference:
Dataset 1: Contradictory, My Dear Watson
Dataset 2: Stanford Natural Language Inference Corpus

Dataset 1 contains 12120 unique examples in the train set. Which consists of premise-hypothesis pairs in **fifteen different languages, including Arabic, Bulgarian, Chinese, German, Greek, English, Spanish, French, Hindi, Russian, Swahili, Thai, Turkish, Urdu, and Vietnamese.**
Here, the premise provides the context with which the hypothesis sentence will be compared with. The Data set also contains information about what language the text is written in along with the class label for each data point.

Dataset 2 contains The SNLI corpus (version 1.0) is a collection of 570k human-written English sentence pairs manually labeled for balanced classification with the labels entailment, contradiction, and neutral, supporting the task of natural language inference (NLI), also known as recognizing textual entailment (RTE). The main aim is to serve both as a benchmark for evaluating representational systems for text, especially including those induced by representation learning methods, as well as a resource for developing NLP models of any kind.

The challenging part was combining both the datasets as dataset 1 had extra columns which had to be dropped and merging was not fruitful as some of the sentences from premises and hypothesis did not match with each other, in other words, some sentence were not translated to english and hence, the sentence did not make complete sense.

4. Data Analysis

4.1. Dataset 1:

The following figure shows the distribution of languages contained in the training set.

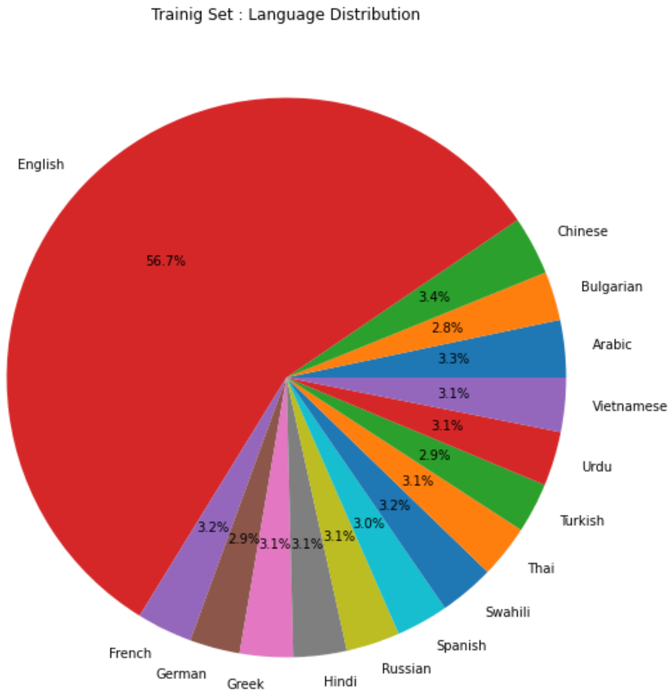
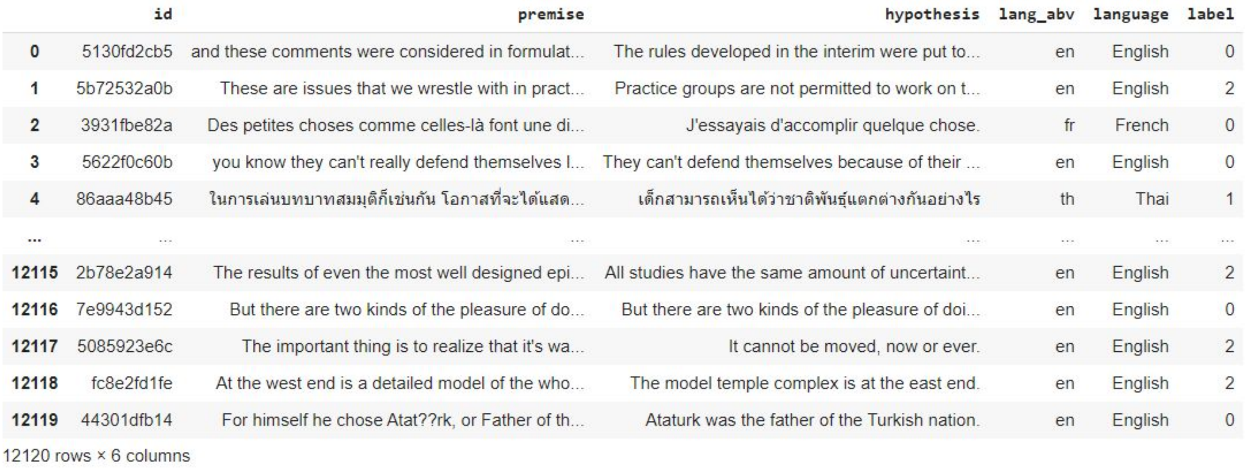


Fig 1: Distribution of Languages

The figure below shows the train set data having the columns id, premise, hypothesis, lang_abv, language and label. The label contains 0, 1, or 2 (corresponding to entailment, neutral, and contradiction) respectively.



	id	premise	hypothesis	lang_abv	language	label
0	5130fd2cb5	and these comments were considered in formulat...	The rules developed in the interim were put to...	en	English	0
1	5b72532a0b	These are issues that we wrestle with in pract...	Practice groups are not permitted to work on t...	en	English	2
2	3931fbe82a	Des petites choses comme celles-là font une di...	J'essayais d'accomplir quelque chose.	fr	French	0
3	5622f0c60b	you know they can't really defend themselves l...	They can't defend themselves because of their ...	en	English	0
4	86aaa48b45	ในการเล่นบทบาทสมมุติก็เช่นกัน โอกาสที่จะได้แสด...	เด็กสามารถเห็นได้ว่าชาติพันธุ์แตกต่างกันอย่างไร	th	Thai	1
...	...	...	...	...	...	...
12115	2b78e2a914	The results of even the most well designed epi...	All studies have the same amount of uncertaint...	en	English	2
12116	7e9943d152	But there are two kinds of the pleasure of do...	But there are two kinds of the pleasure of doi...	en	English	0
12117	5085923e6c	The important thing is to realize that it's wa...	It cannot be moved, now or ever.	en	English	2
12118	fc8e2fd1fe	At the west end is a detailed model of the who...	The model temple complex is at the east end.	en	English	2
12119	44301dfb14	For himself he chose Atat??rk, or Father of th...	Ataturk was the father of the Turkish nation.	en	English	0

12120 rows × 6 columns

Fig 2: Train set data

4.2. Dataset 2:

The second dataset was taken from the SNLi corpus. As shown in the figure below, based on the sentences classified in the labels 0, 1, or 2 (corresponding to entailment, neutral, and contradiction), a plot was generated and for each of the labels, the histogram peaks at about 175000 premise-hypothesis pairs.

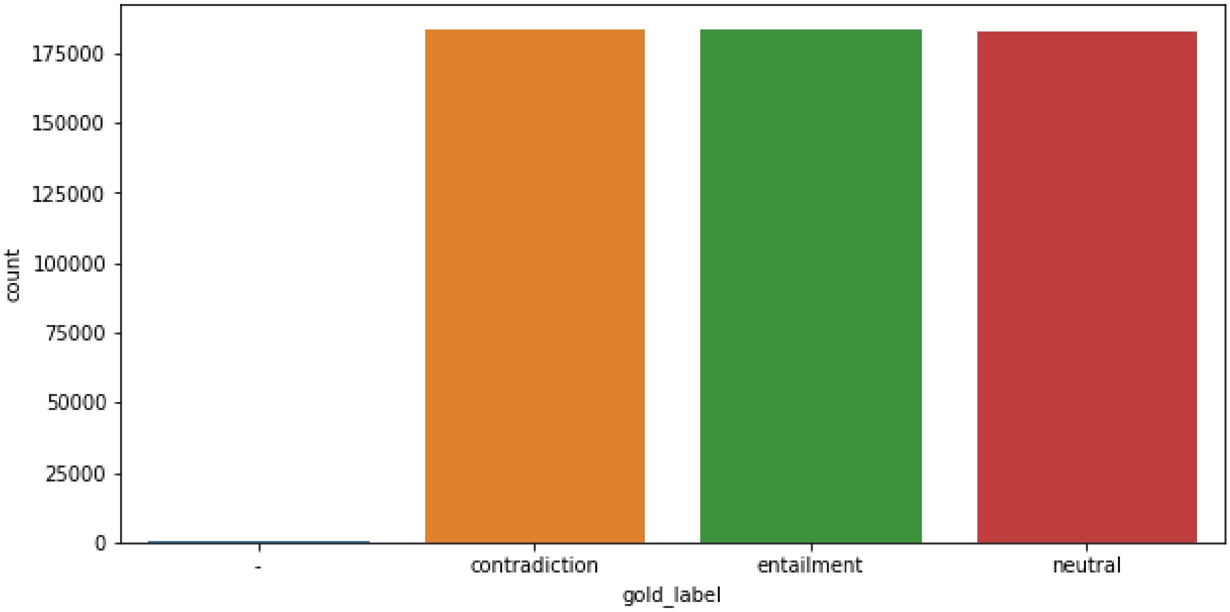


Fig3: Train Set data of dataset 2

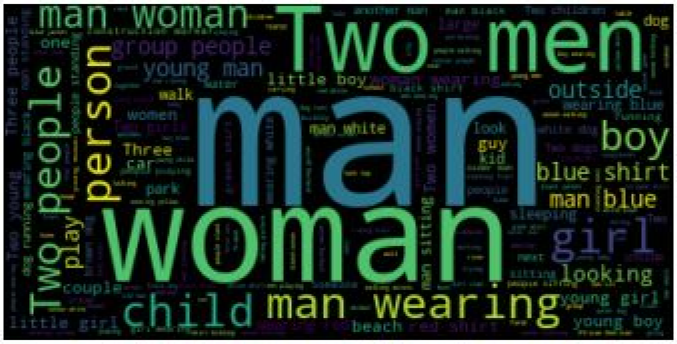


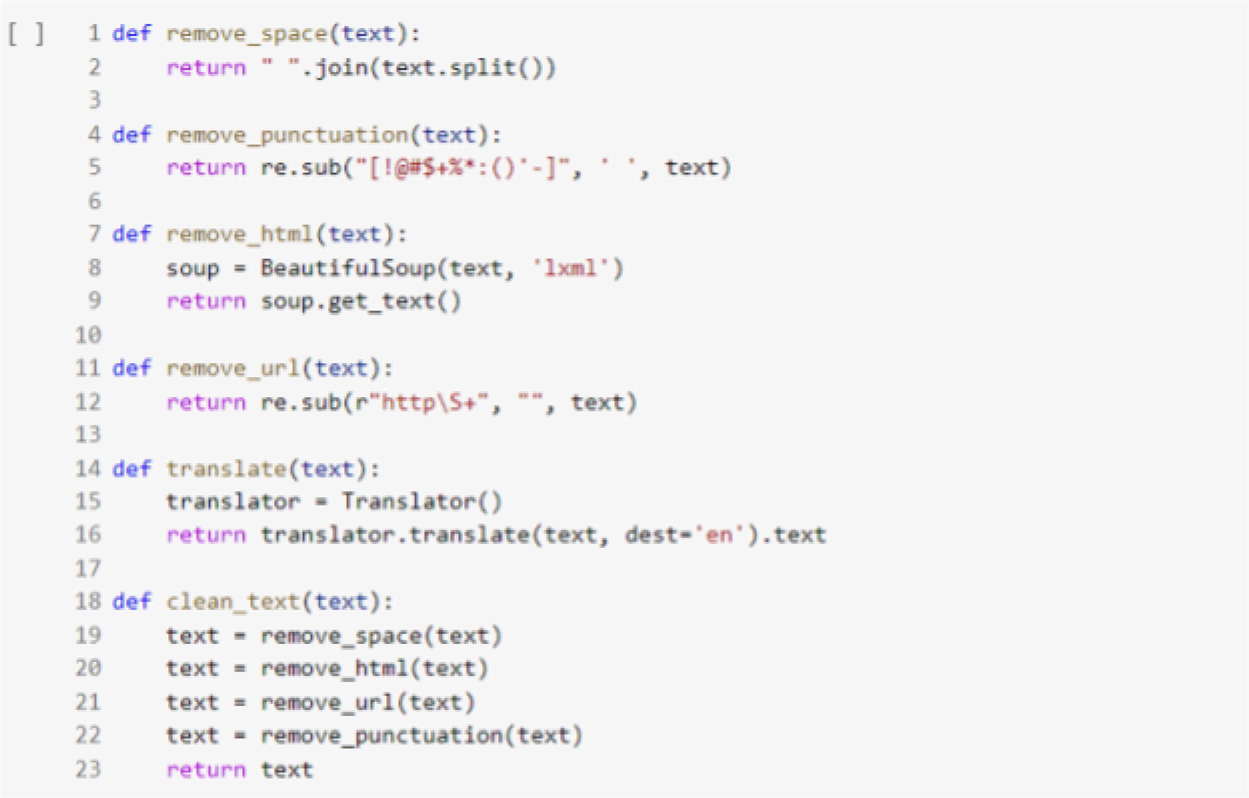
Fig4: Common Words in dataset 2.

The figure above depicts a word cloud of some of the most common words in the first 10000 data points in dataset 2 train set.

5. Preprocessing Dataset:

5.1. Dataset 1:

For dataset 1, we clean the dataset by removing any additional brackets, space, punctuation, html and url. Once the dataset was cleaned, it was then stored in a csv.



```
def remove_space(text):
    return " ".join(text.split())

def remove_punctuation(text):
    return re.sub("[!@#$+%*:()'-]", ' ', text)

def remove_html(text):
    soup = BeautifulSoup(text, 'lxml')
    return soup.get_text()

def remove_url(text):
    return re.sub(r"http\S+", "", text)

def translate(text):
    translator = Translator()
    return translator.translate(text, dest='en').text

def clean_text(text):
    text = remove_space(text)
    text = remove_html(text)
    text = remove_url(text)
    text = remove_punctuation(text)
    return text
```

Fig5: Text clean-up code

5.2. Dataset 2:
For dataset 2, we clean the dataset by removing any additional brackets, space, punctuation, html and url. Once the dataset was cleaned, it was then stored in a csv.

5.3. Combined Dataset:
The dataset 1 and dataset 2 were combined and stored in a csv format. Although we planned this out to help train the model and improve the accuracy, we faced a lot of hurdles when converting all the 15 different languages to english. One of the hurdles we faced was the sentence, taken from premise and hypothesis, when compared did not make complete sense or at times the sentence did not convert to english. Hence, we ruled out the plan of using the combined dataset and went forward with training the network architecture using the dataset 2.

6. Different Neural Network Model Architecture:

Models	Train set accuracy	Validation set accuracy
Simple NN	33.7	33.6
Single LSTM	33	33
Single Bidirectional LSTM	55	37.7
Two Bidirectional LSTM	46.48	39.84
Single GRU	47.30	38.94
Two Bidirectional GRU	45.89	40.14
Three Bidirectional GRU	48.13	40
Single Convolution	37.42	34.35
Simple RNN	48.75	35.48
Simple Bidirectional RNN	44	37.38
MiniVGG	98.96	52.78
BERT	85	75
LSTM+ SimpleRNN	41	35
Triple LSTM and Tokenization(Method 2)	95	47
Double LSTM and	98	42

Tokenization(Method 2)		
Simple Bert	85	75.15
BERT (More Epochs)	98.50	75.60
BERT (Dropout)	99.56	76.50
BERT (Dataset 1) MultiLinguistic	98.72	60.85

From the above model, we can see that Bidirectional Encoder Representations from Transformers (BERT) neural network model gave us the best generalization. Now using this network, we move forward with tuning the hyperparameters.

7. Tuning Hyperparameters:

7.1. Tokenization:

There are two ways of tokenization. They are:

- Dictionary Tokenization for training data only
- Dictionary Tokenization for the whole dataset.

7.2. Optimizers :

We tried using different optimizers and tuned our network model. The optimizers with their respective accuracies can be shown in the following table:

Optimizers	Validation set Accuracy
Adam	76.50
RMSprop	35.50

From the above table, we can see that Adam optimizer gave us the best accuracy out of them. Using this optimizer, we tune the rest of the hyperparameters.

7.3. Dropout Values:

We tried using different Dropout values and tuned our network model. The Dropout values with their respective accuracies can be seen in the following table:

Dropout Values	Validation set Accuracy

0.3	33.05
0.2	76.5
0.1	75.6

From the above table, we can see that 0.2 gave us the best accuracy. Hence we use that value to tune the rest of the hyperparameters.

7.4. Loss functions:
We tried using different loss functions and tuned our network model. The Loss functions with their respective accuracies can be seen in the following table:

Loss Functions	Validation set Accuracy
Binary_cross-entropy	76.40
Categorical_cross-entropy	76.50

From the above table, we can see that Categorical Cross-entropy loss function gave us the best accuracy. Hence we use that value to tune the rest of the hyperparameters.

7. Final Code:

After tuning the hyperparameters from above, a final code was generated and run for final training. A total of 80000 dataset values were used for training and 20000 data points were used for validation. The accuracy obtained was 83 percent for validation set and 99.23 percent for train set for the BERT network model.