Applying NLP Tasks to Diary Book By Using Bert

Deniz Dölek
University of Konstanz

ABSTRACT

The purpose of this project is to extract information from the book "Andy Warhol Diaries" by using state-of-art methods. Natural Language Processing(NLP) deals with how machines can derive meaning from text or speech. This project focuses on NLP tasks, specifically Named Entity Recognition(NER) and Sentiment Analysis(SA) for gathering the necessary information. NER aims to get person, location, organization related information and SA detects the conveyed sentiment of passages which may be positive or negative. These NLP tasks are handled by the transformer-based machine learning technique, BERT, and its more lightweight version, DistilBERT.

KEYWORDS

extract information, NLP, named entity recognition, sentiment analysis, Bert, transformer

1 INTRODUCTION

Natural language processing(NLP) is the study of banding data science and natural languages together. The main goal of NLP is to utilize computers to read, understand and derive meaning from text and speech.It is based upon artificial intelligence, machine learning, and deep learning, and deals with computer science and computational linguistics.

NLP is one of the most significant studies in today's era by not only creating easy ways to connect computers and humans but also affecting so many applications that focused on big data. It offers user-friendly features that make our lives easier such as information extraction, spell checking, machine translation, summarization and sentiment analysis. With the developments in computational power and access to data, NLP achieves important results in areas like healthcare, media, finance, and human resources, among others.[14] Name Entity Recognition and Sentiment Analysis, which are the most popular tasks in NLP, are the main subjects of this project.

Named entity recognition(NER) is one of the most important entity detection methods that helps solving information retrieval, relation extraction, and question answering in NLP systems. NER consists of two significant stages; the first part involves detecting a named entity and the second part concerns itself with putting the entity into one of the predefined categories.[9] Entities are chunks of particular sentences and NER tries to extract information regarding people, organizations, locations, and others from said chunks.1

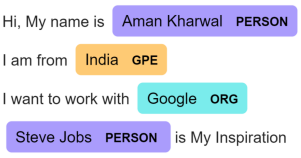


Figure 1: NER example[]

In addition, sorting unstructured data and getting key information from large datasets become easier with the help of NER. Therefore, it is significantly more beneficial for optimizing search engines and content classification algorithms.

In recent years, transformer models and more specifically the BERT model came into play to increase the performance of the NER related tasks.[8] To identify what information is relevant and to classify various entities, a NER model requires training data, for which transformers provide crucial contributions and improvements during the training stage.

Sentiment analysis is one of the most important tasks where NLP methods have been dominantly applied. The aim of Sentiment Analysis is to extract information about personal feelings and emotions from the text. Especially, in business life, Sentiment Analysis plays a crucial part in understanding customer behavior and needs.[13] From reviews, emails, or even tweets, it provides important feedback from a large amount of data in a way that is not possible when observed manually. In Figure 2, the review passed on a transformer firstly, then applied a classifier to find out the targeted sentiment.

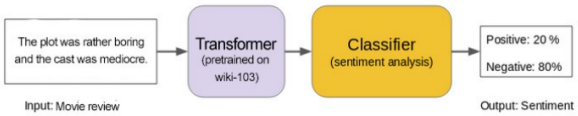


Figure 2: Sentiment Analysis Example[3]

NLP areas such as NER and Sentiment Analysis, stepped into the new age after the concept of Transformers were introduced by Google in 2018. Before transformers, sequence-to-sequence models such as Recurrent Neural Networks (RNNs) and Long-Short-Term-Memory (LSTM) networks were shown to have some problems that negatively effect performance. For RNN, the problem was vanishing gradients caused by long-term memory loss. This problem was overcome with LSTM by including memory cells so that it can maintain long term temporal dependencies. Nonetheless, the main problem for both models was that they take a single input at a time and can not run in parallel. At this point, transformers got involved and revolutionize the process, explained in the paper "Attention Is All You Need". The solution that they introduced is the additional training parallelization and the attention mechanism. Parallelization allows for training on large datasets over multiple times and the attention-mechanism steps in to solve the vanishing gradient problem by calculating all weights for each word.

In their words, "Self-Attention" was described as "an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence". In Figure 3, we can observe that with the help of self-attention, specific words can be more understandable by looking at other words in the input sequence. "The" is connected to "animal" and they both refer to "it".

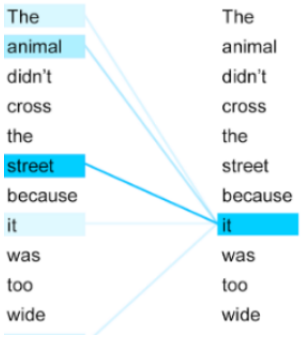


Figure 3: Self-attention example[12]

2 RELATED WORKS

2.1 BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Authors introduce a new language model that is based on transformers, Bidirectional Encoder Representations, briefly named BERT. It provides new state-of-the-art results on several NLP tasks. The paper describes bidirectional pre-training models for language representations and shows how it works in both directions. BERT deploys a fine-tuning-based approach which means pretraining a model with a large unlabeled dataset, followed by a fine-tuning phase, using a smaller labeled dataset. BERT has two important features which can be summarized as: "masked language model" and "next sentence prediction".

One of the most important characteristics that distinguishes BERT from other approaches is Input/Output representations. WordPiece embeddings are used with a 30,000 token vocabulary, which involves special tokens such as [CLS] and [SEP]. Input representation allows for attaining more information about corresponding tokens, segments, and position embeddings.

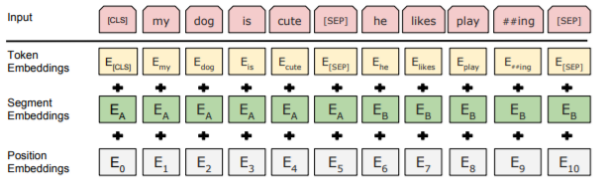


Figure 4: Token representation [6]

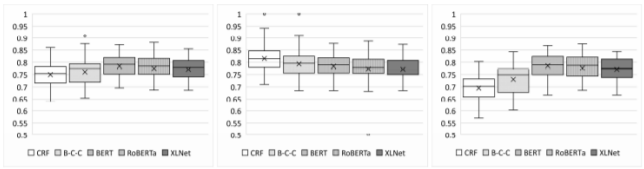
Experiments prove that bidirectional architectures and pretrained models achieve great results for a wide range of NLP tasks. Especially BERTLarge, an extensive model built upon BERT which consists of 24 layers, 1024 hidden dimensions, 16 attention heads and 335 million parameters performs operations for both fine-tuning and feature-based approaches at a competitive level with an efficacy that rivals or exceeds other state-of-the-art methods. The results are shown in Figure9 [6]

2.2 Evaluating Pretrained Transformer-based Models on the Task of Fine-Grained Named Entity Recognition

Authors describe the progress of NER throughout time and show how different approaches affect performance. Fine-Grained Named Entity Recognition is a subtask of NER and objectives are the same but the number of entity types is higher. Main contribution of their work is that it shows a comparison between transformer-based models (BERT, RoBERTa, and XLNet) and non-transformer-based models (CRF and BiLSTM-CNN-CRF), hence making it a valuable source to confer for direction. Another important finding touched upon was that by taking f1 scores into consideration, they proved that transformer-based models are better than non-transformer-based models. [8]

System	Dev F1	Test F1
ELMo (Peters et al., 2018a)	95.7	92.2
CVT (Clark et al., 2018)	-	92.6
CSE (Akbik et al., 2018)	-	**93.1**
Fine-tuning approach		
$\text{BERT}_{\text{LARGE}}$	96.6	92.8
$\text{BERT}_{\text{BASE}}$	96.4	92.4
Feature-based approach ($\text{BERT}_{\text{BASE}}$)		
Embeddings	91.0	-
Second-to-Last Hidden	95.6	-
Last Hidden	94.9	-
Weighted Sum Last Four Hidden	95.9	-
Concat Last Four Hidden	96.1	-
Weighted Sum All 12 Layers	95.5	-

Figure 5: CoNLL-2003 Named Entity Recognition results



(a) Distribution of F1 scores (b) Distribution of Precision (c) Distribution of Recall

Figure 6: Distribution the performance of the five models used

2.3 Aspect-Based Sentiment Analysis Using BERT

Traditional sentiment analysis task is moved up into a higher level with this paper and the authors consider not only sentiment but also aspects of the context. Aspect-based sentiment analysis (ABSA) is more complicated than the standard version and offers a combined model that includes aspect classification and sentiment classification. It takes advantage of BERT and uses the sentence pair classifier model. The Sentence Pair Classifier task helps to understand the semantic connection between two sentences.

The reason for choosing this paper is it widened my horizon for future work. I used the standard Sentiment analysis for the tasks and this paper shows how can I improve my work by adding an aspect classifier. This classifier basically identifies the text as "related" or "unrelated" unlikely to the sentiment classifier which labels 'positive' or 'negative'.They observe that the combined method gives better results than aspect based sentiment classification[1]

2.4 Analyzing ELMo and DistilBERT on Socio-political News Classification

This study shows us the comparison between two important two state-of-the-art language models, ELMo and DistilBERT. The purpose of the paper is to observe how much can these models be benefited from nearly without any changes to the pretraining outputs. Even DistilBERT is smaller than ELMo, it has close performance according to F-score. Also, DistilBERT has a big advantage in training time, it is faster at the rate of 83

Considering my project and choosing DistilBERT for sentiment analysis, I noticed that DistilBERT is commonly used in classification tasks and this paper proves that how it can be helpful and how effective it is. [4]

3 METHODOLOGY

3.1 BERT

BERT is a powerful language model that initiates a new era in NLP. In Section 2.1 BERT is introduced conservatively according to its paper. Now, some important details about BERT and related information for this project will be described in this section. First of all, BERT is a pre-trained model that means it is trained in a large dataset before you use it and allows you to finetune on a specific task, in this project it is NER.

The model "bert-base-cased model" is used for the named entity recognition tasks for this project. bert-base-NER is a fine-tuned BERT model that consists of 12-layer, 768-hidden, 12-heads, 109M parameters. It was fine-tuned on the English version of the standard CoNLL-2003 Named Entity Recognition dataset. Reuters news stories constitute this dataset which is the standard version for the tags used in train sets to train NER models. In Figure 3, we can see 12 layers can work together and this parallelization gives opportunity to handle large datasets.

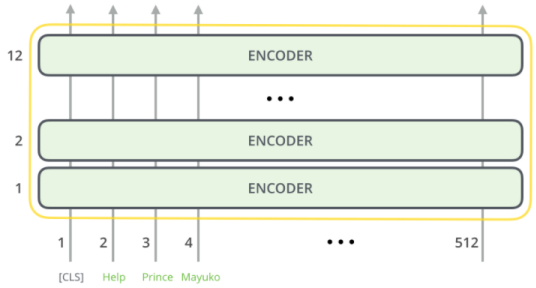


Figure 7: Illustration of BERT model with 12 layers and 512 tokens[2]

BERT can finetune each NLP task easily. If we focus on NER, the model gets a text sequence and each sequence is tagged with types of entities such as person, organization, etc. After training the model with the output embeddings of each word with the tags into a classification layer, the model will have the ability for predicting NER labels.

"Masked Language Model" and "Next Sentence Prediction" are two important concepts that BERT introduced. These two concepts play a crucial role in the improvement of the NER operations. Unlikely old versions such as GloVe and word2vec, words become more dimensional and easier to understand for the model. Furthermore, a self-attention mechanism is used for the first time with BERT. It also provides to increases the depth of meanings for the words and becomes easier to identify their labels for NER models.

"masked language model(MLM)" is a new language model that is based on masking a word in a sequence of inputs and then tries to find out that word. The special [MASK] token substitutes with a word with 15% randomly. This method is beneficial for finding side meanings of the words. The model is not only checks the words left-to-right or right-to-left, it concentrates the whole sequence for deriving meaning.

Example:

Input: "I have played this video [MASK] and it has got FPS problem."

Output: "I have played this video game and it has got FPS problem."

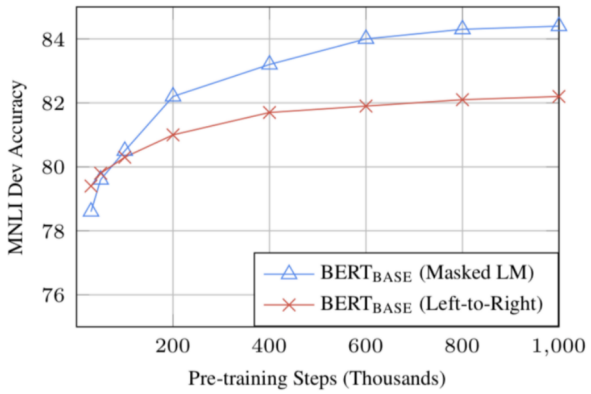


Figure 8: Comparison between Masked LM and left-to-right approach[6]

In Figure 8, since only 15% of words are predicted in each batch, learning starts slowly but outperforms the left-to-right approach in the progress of time.

"Next Sentence Prediction (NSP)" tries to establish a connection between the sentences and find out their relation. Instead of MLM's masked words, NSP tags the sequence of the sentence as isNext or is notNext and predict the following sentences by focusing the whole context of the text.

Example

Sentence A : [CLS] The woman went to the supermarket. [SEP]

Sentence B : He bought vegetables. [SEP]

Label : IsNext

In the training phase, half of the time sentence B is correctly used as the next sentence, and the other half, a random sentence is selected from the text. This provides that the model can challenge with different sentences to find the right one and makes it easier to understand the context of the text.

The mechanism of self-attention is a core module for transformers-based models. Introduction of Multi-head attention with the transformers, attention mechanism improves its calculation and gives better results. Self-attention derives the meanings by looking the other words and avoid ambiguity.

The most common example for understanding self-attention is "I arrived at the bank after crossing the _____" For deciding what refers "bank" means a river or a financial institute, we need to know how the sentence is ending, and also self-attention.

3.2 DistilBERT

DistilBERT, which is developed by Huggingface, is the smaller and lighter version of Bert and it is built on the same general architecture. It is 40% smaller and 60% faster but despite all, it keeps 97% score of the language understanding capabilities. [11] DistilBERT is aimed to reduce the large size of the BERT by using a method named knowledge distillation, which means that a smaller model is trained based on loss of the original model.[11]

Model	Score	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	WNLI
ELMo	68.7	44.1	68.6	76.6	71.1	86.2	53.4	91.5	70.4	56.3
BERT-base	79.5	56.3	86.7	88.6	91.8	89.6	69.3	92.7	89.0	53.5
DistilBERT	77.0	51.3	82.2	87.5	89.2	88.5	59.9	91.3	86.9	56.3

Figure 9: Comparison on the dev sets of the GLUE benchmark

In this project, DistilBERT base uncased finetuned SST-2 was used for Sentiment Analysis. This model has an accuracy of 91.3 on the development set.11 SST-2 stands for Stanford Sentiment Treebank version 2 and it is a dataset that contains 215,154 phrases with fine-grained sentiment labels in the parse trees of 11,855 sentences from movie reviews.

Fine-tuning hyper-parameters for this model
learning_rate = 1e-5
batch_size = 32
warmup = 600
max_seq_length = 128
num_train_epochs = 3.0

3.3 Simple Transformers

Simple Transformers library, which is created by Thilina Rajapakse, provides easy and simple usage for transformers. It is created as a wrapper around the excellent Transformers library by Hugging Face in his words.[10] It uses Pytorch at the backend. The library provides a number of pre-trained models such as BERT, RoBERTa and helps to solve many of the NLP tasks, mainly text classification, NER,and question answering. It involves three main steps: Initialize a model which depends on what NLP tasks you deal with, train the model, and evaluate it.

In this project, NER operations are handled by the Simple Transformers library and “bert-base-cased model” is selected for NERModel as mentioned before. The model is using the default list of labels from the CoNLL dataset which uses the following tags classically.[7]

["O", "B-MISC", "I-MISC", "B-PER", "I-PER", "B-ORG", "I-ORG", "B-LOC", "I-LOC"]

The IOB format is the standard way of tagging tokens for chunk structures in computational linguistics. IOB stands for inside, outside, beginning. The prefixes come before a tag (e.g. I-PER) and indicate the position of the word. The B-prefix shows that the tag is in the beginning chunk, the I-prefix means for tags that it is inside the chunk, and the O-prefix indicates the tag as it is outside any chunk.

3.4 CUDA

CUDA, is an API for developers, that was created by NVIDIA. It reduces significant time for computation by providing computing power and allowing to use GPUs. A GPU can be much powerful at computing than a CPU in some specific tasks.[5] In deep learning, parellization is very significant for computation and the architecture of neural networks as well as BERT is suitable for GPUs which are beneficial for parallel computing. GPUs have high-bandwidth memories and this makes them faster.

3.5 Google Colab

“Colab” which is shortened from collaboratory is developed by Google. Users can implement and execute python code through the browser with the help of Colab. This project deployed in Colab Pro and the hardware details can be seen in Figure 10.

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 470.63.01    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   39C    P8     9W /  70W |      0MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Figure 10: Details of Hardware

4 TRAINING PART FOR NER

Named Entity Recognition tasks are solved with the BERT model and simpletransformers library. For the dataset, I choose ner_dataset which is used commonly for this specific task. It involves 1M x 4 dimensions and the columns are ’Sentence’, ’Word’, ’POS’, ’Tag’.(Ref2*) The representation can be seen in Figure 12 The data is preprocessed for optimizing simpletransformers’ NERModel function.

The number of words in the dataset is 35178 and the labels that belong to words are person, geographical entity, organization, geopolitical entity, time indicator, artifact, event, and natural phenomenon. Our dataset mostly contains words related to geographical locations, geopolitical entities, and person names. The distribution of the labels in the dataset is shown in Figure 11.

Fine-tuning hyper-parameters are changed in this task:
num_train_epochs = 2z
learning_rate = 5e-5
batch_size = 64
eval_size = 64
max_seq_length = 128
num_train_epochs = 3.0

As mentioned before, ’bert-base-cased’ is used and the hyperparameters are taken from a work that examines optimal parameters for BERT.(Ref*) After training and evaluation phase, the results are:
’eval_loss’: 0.17
’precision’: 0.82
’recall’: 0.75
’f1_score’: 0.78

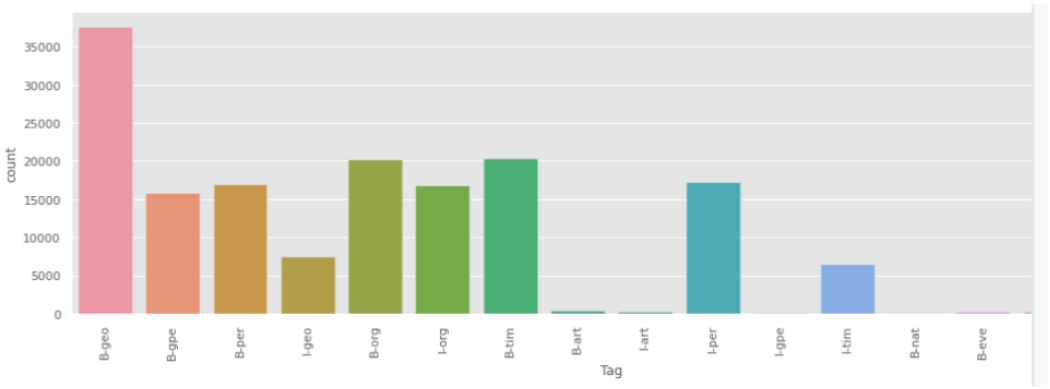


Figure 11: The distribution of the labels

	Sentence #	Word	POS	Tag
0	Sentence: 1	Thousands	NNS	O
1	Sentence: 1	of	IN	O
2	Sentence: 1	demonstrators	NNS	O
3	Sentence: 1	have	VBP	O
4	Sentence: 1	marched	VBN	O
5	Sentence: 1	through	IN	O
6	Sentence: 1	London	NNP	B-geo
7	Sentence: 1	to	TO	O
8	Sentence: 1	protest	VB	O
9	Sentence: 1	the	DT	O
10	Sentence: 1	war	NN	O
11	Sentence: 1	in	IN	O
12	Sentence: 1	Iraq	NNP	B-geo
13	Sentence: 1	and	CC	O
14	Sentence: 1	demand	VB	O
15	Sentence: 1	the	DT	O
16	Sentence: 1	withdrawal	NN	O
17	Sentence: 1	of	IN	O
18	Sentence: 1	British	JJ	B-gpe
19	Sentence: 1	troops	NNS	O
20	Sentence: 1	from	IN	O
21	Sentence: 1	that	DT	O
22	Sentence: 1	country	NN	O
23	Sentence: 1	.	.	O

Figure 12: Example of a sentence representation

```
[{'Went': 'O'},
 {'to': 'O'},
 {"Halston's": 'I-PER'},
 {'birthday': 'O'},
 {'dinner': 'O'},
 {'for': 'O'},
 {'Victor': 'B-PER'},
 {'at': 'O'},
 {"Pearl's": 'B-GEO'},
 {'he': 'O'},
 {'did': 'O'},
 {'not': 'O'},
 {'want': 'O'},
 {'to': 'O'},
 {'do': 'O'},
 {'a': 'O'},
 {'big': 'O'},
 {'thing': 'O'},
 {'at': 'O'},
 {'the': 'O'},
 {'house.': 'O'}]
```

Figure 13: The first predicted sentence with labels

4.1 Prediction

The book is in diary format and it involves 2246989 words in 2024 days. The text of the book splits into sentences and the predict function is applied to each sentence. In Figure 13, the first sentences of the book are shown with the labels.

If we look more specifically, there are 5371 unique people in the book. The most commons are “Bob”, “Fred” and “John”. By looking at their index, we can make inferences about their existence that can be an early stage of the book or remains in the whole story. Also, we can separate the persons for each day. However, applying predict function separately for each day is a very time-consuming process. New solutions are needed for effectiveness in daily NER as future work.

Furthermore, the most common locations are New York, LA, California in Andy’s diary. He mentioned, “night” more than “mornings and “afternoons”. In the book the count of each geopolitical entity such as “American”, “English”, “Italian” are very close.

5 SENTIMENT ANALYSIS FOR DIARY

In this project, the pipeline which is created by Hugginface is used for SA. The tokenization and the model in the pipeline, are using “distilbert-base-uncased-finetuned-sst-2-english”. SST2 stands for The Stanford Sentiment Treebank and involves 215,154 phrases with fine-grained sentiment labels in 11,855 sentences from movie reviews. Our model is classifying sequences according to positive or negative sentiments.

After applying the model for each day, we can observe that negative days are well ahead of positives days. Just one final detail, the model takes 512 tokens but in some days the words are more than 512. The problem is solved with the truncation method. After 512 tokens it splits the text and continues to calculate the remaining part.

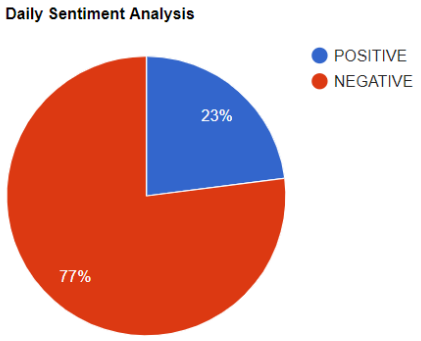


Figure 14: SA results for all days

6 DISCUSSION

BERT and DistilBERT procured good results for NLP tasks. NER is achieved with %82 f1-score. This can be improved by adding new technologies, CRF decoder, or applying cross-validation, a different softmax layer in future work.

NER is not only focusing on extracting persons but also different entity types such as location and geopolitical information. However, in this project, the labels which are natural phenomena and organizations can not achieve very well.

One of the problems that I faced is there was no ground-truth data. Therefore, I can not compare the entities that I extracted from the book.

In SA part, the model only targeted two emotions. In future work, the model should be improved by focusing more on specific emotions and different moods. Also, different approaches such as aspect-based should be considered in development.

Since I have lack hardware, I decided to use Google Colab that allows you to work with their GPUs. However, the models of GPUs can change in each accession and this affects training times every time. Also, before using the Pro version of Colab, so many problems occurred.

7 CONCLUSION

In conclusion, this project aims to apply state-of-art models such as BERT and DistilBERT to a diary book. Information extraction from texts becomes easier with the improvement of Natural Language Processing. Since transformers enter this area, NLP tasks significantly increased their performances. In this project, the NLP tasks, which are Named Entity Recognition and Sentiment Analysis, achieved good results. The new technologies and open-sourced libraries are examined to improve the results.

REFERENCES

[1] Mohammed M. Abdelgwad. 2021. Arabic aspect based sentiment analysis using BERT. *CoRR* abs/2107.13290 (2021). arXiv:2107.13290 https://arxiv.org/abs/2107.13290

[2] Jay Alammar. The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning). https://jalammar.github.io/illustrated-bert/. (????).

[3] Oliver Atanaszov. Transformer Fine-Tuning for Sentiment Analysis. https://ben0it8.github.io/transformer-finetuning/. (????).

[4] Berfu Büyüköz, Ali Hürriyetoğlu, and Arzucan Özgür. 2020. Analyzing ELMo and DistilBERT on Socio-political News Classification. In *Proceedings of the Workshop on Automated Extraction of Socio-political Events from News 2020*. European Language Resources Association (ELRA), Marseille, France, 9–18. https://aclanthology.org/2020.aespen-1.4

[5] deeplizard. CUDA Explained - Why Deep Learning Uses GPUs. https://deeplizard.com/learn/video/6stDhEA0wFQ. (????).

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR* abs/1810.04805 (2018). arXiv:1810.04805 http://arxiv.org/abs/1810.04805

[7] huggingface. Dataset Card for "conll2003". https://huggingface.co/datasets/conll2003. (????).

[8] Cedric Lothritz, Kevin Allix, Lisa Veiber, Tegawendé F. Bissyandé, and Jacques Klein. 2020. Evaluating Pretrained Transformer-based Models on the Task of Fine-Grained Named Entity Recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*. International Committee on Computational Linguistics, Barcelona, Spain (Online), 3750–3760. https://doi.org/10.18653/v1/2020.coling-main.334

[9] Christopher Marshall. What is named entity recognition (NER) and how can I use it? https://medium.com/mysuperai/what-is-named-entity-recognition-ner-and-how-can-i-use-it-2b68cf6f545d. (????).

[10] Thilina Rajapakse. simpletransformers. https://github.com/ThilinaRajapakse/simpletransformers. (????).

[11] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[12] Ashish Vaswani, Samy Bengio, Eugene Brevdo, François Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Lukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. 2018. Tensor2Tensor for Neural Machine Translation. *CoRR* abs/1803.07416 (2018). arXiv:1803.07416 http://arxiv.org/abs/1803.07416

[13] Christian Versloot. How to perform Sentiment Analysis with Python, HuggingFace Transformers and Machine Learning. https://www.machinecurve.com/index.php/2020/12/23/easy-sentiment-analysis-with-machine-learning-and-huggingface-transformers/. (????).

[14] Diego Lopez Yse. Transformer Fine-Tuning for Sentiment Analysis. https://ben0it8.github.io/transformer-finetuning/. (????).

(Ref*) Jack Morris - Does Model Size Matter? A Comparison of BERT and DistilBERT (Ref2*) https://www.kaggle.com/namanj27/ner-dataset