VIETNAM GENERAL CONFEDERATION OF LABOR **TON ĐUC THANG UNIVERSITY FACULTY OF INFORMATION TECHNOLOGY**

FINAL PROJECT

KNOWLEDGE DISCOVERY AND DATA MINING

NATURAL LANGUAGE PROCESSING WITH DISASTER TWEETS

Instructor: **ThS. HOÀNG ANH**

Student: **HUỲNH NHẬT BẢO – 520H0605**

Course **:** **24**

HO CHI MINH CITY, 2024

VIETNAM GENERAL CONFEDERATION OF LABOR **TON ĐUC THANG UNIVERSITY FACULTY OF INFORMATION TECHNOLOGY**

FINAL PROJECT

KNOWLEDGE DISCOVERY AND DATA MINING

NATURAL LANGUAGE PROCESSING WITH DISASTER TWEETS

Instructor: **ThS. HOÀNG ANH**

Student: **HUỲNH NHẬT BẢO – 520H0605**

Course **:** **24**

HO CHI MINH CITY, 2024

THANKFUL WORD

First of all, I and my team was able to complete the report in the most comprehensive and effective manner after working for a semester with the enthusiastic aid and support of ThS. Hoàng Anh. His instruction has provided our pupils with a wealth of knowledge as well as comprehensive abilities in the particular area. Although a three of months is a short period, it has allowed me to tackle the major step by step with a strong foundation, especially with the support and assistance of seasoned lecturer like my instruction. We thank you a lot for all that you have taught us and wish you a lot of health!

FINAL ESSAY COMPLETED AT TON DUC THANG UNIVERSITY

I hereby declare that this is my our report and is under the guidance of ThS. Hoàng Anh. The research contents and results in this topic are honest and have not been published in any form before. The data in the tables for analysis, comments and evaluation are collected by the author himself from different sources, clearly stated in the reference section.

In addition, the project also uses a number of comments, assessments as well as data of other authors, other agencies and organizations, with citations and source annotations.

If I find any fraud I take full responsibility for the content of my report. Ton Duc Thang University is not related to copyright and copyright violations caused by me during the implementation process (if any).

Ho Chi Minh city, 15 May, 2024

Author

(sign and write full name)

Bảo

TEACHER'S CONFIRMATION AND ASSESSMENT SECTION

Confirmation section of the instructors

_____

_____

_____

_____

_____

_____

_____

Ho Chi Minh city, day month year
(sign and write full name)

The evaluation part of the lecturer marks the report

_____

_____

_____

_____

_____

_____

_____

Ho Chi Minh city, day month year
(sign and write full name)

SUMMARY

In this Final project, I applied the knowledge I learned from school and self-study such as how to use DistilBERT for analysis, visualize and predict, the ability to work with data modeling in training data, tuning hyperparameter, Performing statistical analysis, predictive modeling, and Generating actionable insights from data analysis to address business challenges, providing evidence-based decision support.

In this report I divide into 3 main parts:

- Chapter 1: Introducing the topic and understanding the datasets.
- Chapter 2: Overview of Keras NLP.
- Chapter 3: Introduce BERT and DistilBERT model use to training.
- Chapter 4: Expland and demo code.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

NLP	Natural Language Processing
BERT	Bidirectional Representations from Transformers
DistilBERT	Distillation Bidirectional Representations from Transformers

LIST OF DIAGRAMS, CHARTS AND TABLES

List of picture

CHAPTER 1. TOPIC INTRODUCTION

1.1 Topic description

This is a topic taken from a competition on Kaggle about natural language processing. In this topic we will process tweets on the social network Twitter. Twitter has become an important communication channel in times of emergency. The ubiquitousness of smartphones enables people to announce an emergency they're observing in real-time. Because of this, more agencies are interested in programatically monitoring Twitter (i.e. disaster relief organizations and news agencies). The ubiquitousness of smartphones enables people to announce an emergency they're observing in real-time. Because of this, more agencies are interested in programatically monitoring Twitter (i.e. disaster relief organizations and news agencies). But, it's not always clear whether a person's words are actually announcing a disaster.

1.2 Topic objective

In this competition, you're challenged to build a machine learning model that predicts which Tweets are about real disasters and which one's aren't.

1.3 Research subjects

The research object is 10,000 tweets collected from the social network Twitter.

1.4 Research scope

The scope of the research is limited to real data posted on Twitter.com website.

1.5 Research Approaches

The research topic uses the DistilBERT machine learning model to analyze and visualize data and expect results and therefore the actual results will score the model.

1.6 Data Specification

The contest provides 3 datasets: train.csv, test.csv, sample_submission.csv which are training set, test set and a sample submission file in the correct format. dataset

includes columns id, text, location, keyword and target, in which the target column only exists in the train.csv file.

CHAPTER 2. OVERVIEW KERAS NLP (Natural Language Processing)

KerasNLP is a library that natively supports language processing throughout the entire development process. This library is an extension of the core Keras API, all high-level modules are multi-layered or models.

KerasNLP uses Keras 3 to work with any TensorFlow, Pytorch or Jax. In our article we use Tensorflow backend to train our models.

CHAPTER 3. OVERVIEW DISTILBERT

3.1 Introduction about BERT

3.1.1 Overview

BERT(Bidirectional Representations from Transformers) is a natural language model developed by Google and announced in 2018. This is one of the most important advances in the field of natural language processing (NLP) in recent years.

3.1.2 Important of BERT

- Revolutionizing NLP: BERT has revolutionized the approach to NLP problems by using models that are pre-trained on big data and then fine-tuned on specific tasks.
- Opening a new era for NLP research and application: After BERT, many similar models such as GPT (Generative Pre-trained Transformer), RoBERTa, and T5 have been developed, further improving and expanding the capabilities of NLP. language models.

3.1.3 Basis Concept

- BERT is a model based on the Transformer architecture, specifically the Transformer encoder.
- The goal of BERT is to create semantic representations of words in their context by learning to understand the meaning of the word in both directions (left and right) of the sentence.

3.1.4 Example

“The bank can guarantee deposits will be safe”.

“He can fish from the river bank”.

- Before BERT:

Previous models such as Word2Vec and GloVe generated a single semantic representation for the word "bank", regardless of context within the sentence. This makes it difficult for the model to accurately determine the meaning of words when encountering different contexts.

- Using BERT:

BERT solves this problem by creating contextual semantic representations. By using the Attention mechanism in both directions, BERT can capture information from surrounding words (before and after the word "bank") to determine its exact meaning.

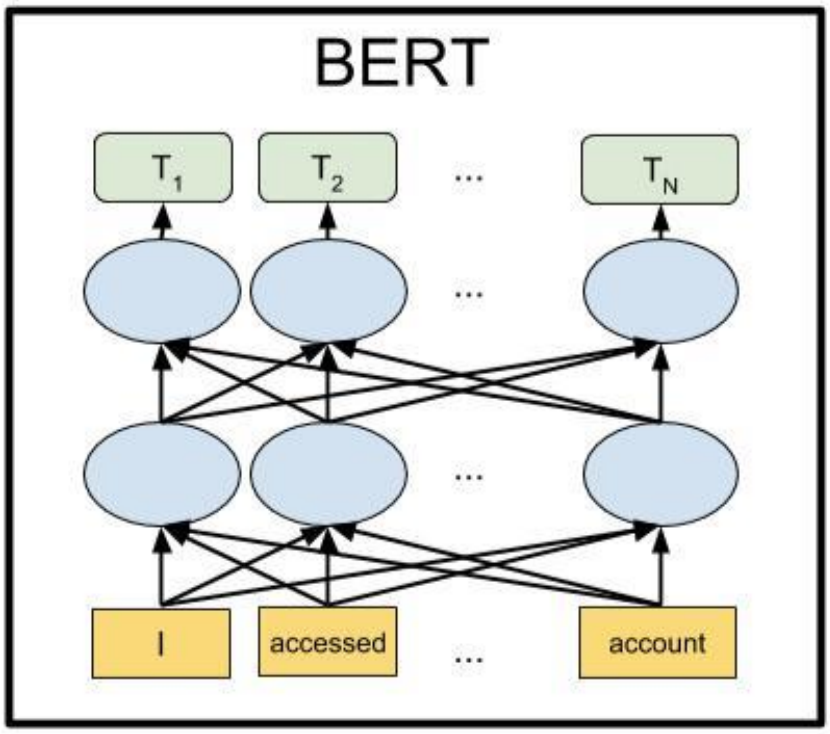
In sentence 1: BERT will look at words like "guarantee", "deposits", and "safe" to understand that "bank" here is a bank.

In sentence 2: BERT will consider words like "fish", "river" to understand that "bank" here is the river bank.

In this way, BERT can generate different semantic representations for the word "bank" based on its context in the sentence.

3.1.5 BERT architecture

BERT will cut the transformer in half and only take the Encoder part on the left and remove the Decoder part on the right, leaving only the part where the text sentence is inserted and the output is the encoder output as shown:



Picture 1: BERT architecture

3.1.6 Pre-training

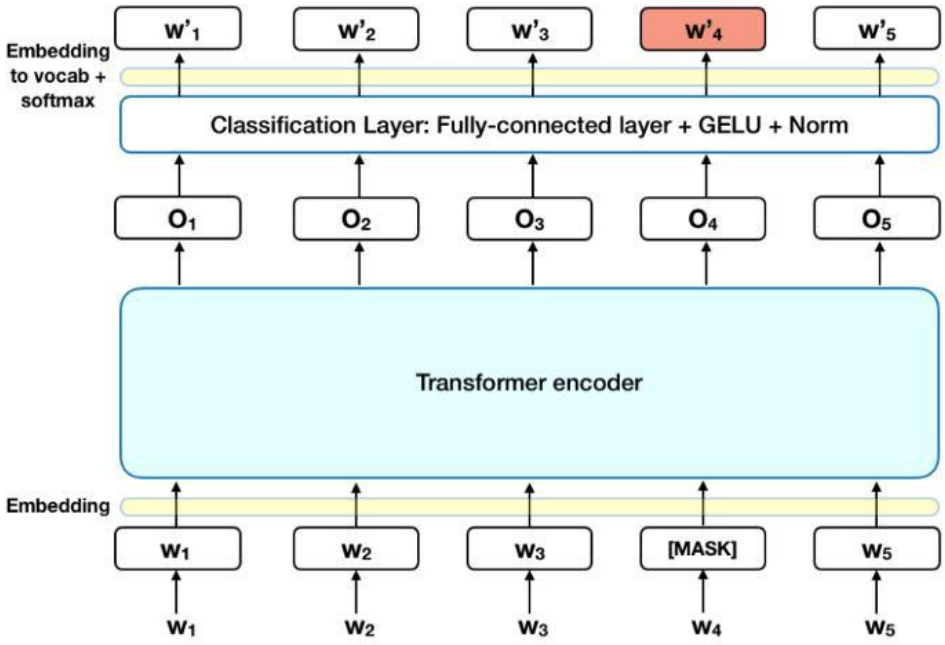
BERT uses two main tasks to train the model:

3.1.6.1 Masked Language Model (MLM)

During the training process, some words in the sentence are randomly masked and the model must guess that masked word based on the context of surrounding words. For example:

Original sentence: "The [MASK] can guarantee deposits will be safe."

BERT predicts the hidden word "bank" based on the context of other words.



Picture 2:Masked Language Model

Detail:

- Add a classification layer on top of the encoder output.
- Bring the vectors in the output encoder to a vector equal to the vocab size, then softmax to select the corresponding word at each position in the sentence.
- Loss will be calculated at the masked position and ignore other positions (to evaluate how correctly/wrongly the model predicts the masked word without other related words).

3.1.6.2 Next Sentence Prediction (NSP)

The model is given two sentences and must be correct determines whether the second sentence is the next sentence in the original text. This helps BERT understand the relationships between sentences and the broader context.

Sentence 1: “He went to the river”.

Sentence 2: “He can fish from the river bank”.

BERT must determine whether sentence 2 is a continuation of sentence 1.

Specific training methods:

- Step 1: Combine 2 sentences together and add some special tokens to separate the sentences. Token [CLS] added at the beginning of the bridge First, token [SEP] added at the end of each sentence. For example, combining two sentences "He went to the river" and "He can fish from the river bank" will become [CLS] He went to the river[SEP] He can fish from the river bank[SEP].
- Step 2: Each token in the sentence will be added with a vector called Sentence Embedding, which actually marks whether the word belongs to the first or second sentence. For example, if it belongs to the first sentence, add a vector of all numbers "0" with the size of Word Embedding (d), and if it belongs to the second sentence, add a vector of all numbers "1".
- Step 3: Then the words in the compound sentence will have a Positional Encoding vector added to mark the position of each word in the compound sentence (if you don't know, please review the article about Transformer).
- Step 4: Insert the chain after step 3 into the network.
- [CLS] Today+P0+S0 today+P1+S0 I+P2+S0 go+P3+S0 to study+P4+S0 [SEP] Study+P5+S1 at+P6+S1 school+P7+S1 very+P8+ S1 fun+P9+S1 [SEP]
- Step 5: Get the encoder output at the token position [CLS] which is transformed into a vector with 2 elements [c1 c2].

- Step 6: Calculate softmax on that vector and output the probability of 2 classes: Following and Not following. To show whether the second sentence follows the first sentence or not, we take argmax.
- Suppose vector [c1, c2] after transform is [2.0,1.0]
- Calculate softmax on [2.0, 1.0]

3.1.7 Pros and cons

Pros

- Comprehensive context:

BERT uses a bidirectional approach, meaning it looks at context from both sides (left and right) of a word. This helps the model better understand the context and meaning of words in sentences.

- High performance on many NLP tasks:

BERT has set new performance records on various NLP tasks such as text classification, entity recognition (NER), question answering, and semantic analysis.

- Flexible fine-tuning capabilities:

BERT can be fine-tuned on specific data sets for specific tasks,

Cons

- Requires large computational resources:

Pre-training BERT requires a large amount of computational resources, including powerful hardware (GPU/TPU) and long training time. This makes training BERT from scratch impractical for many small organizations.

- Large model size:

BERT has many parameters (110 million for BERT-base and 340 million for BERT-large), leading to large memory needs and longer inference times. This can make it difficult to deploy the model in resource-constrained environments.

- Complexity in implementation:

Implementing BERT requires a deep understanding of the Transformer

helping it adapt well and achieve high performance on different problems.

- Powerful pre-training:

The model is pre-trained on a large amount of text data with diverse tasks such as Masked Language Model (MLM) and Next Sentence Prediction (NSP), helping BERT capture many useful linguistic features.

architecture and model optimization techniques, increasing complexity and cost of implementation.

- Optimization and parameter adjustment:

Fine-tuning BERT requires tuning many different parameters and can take a long time to achieve the best performance on specific tasks. This process can be complex and requires thorough testing.

- Performance degrades on uncommon languages:

Although BERT has been pre-trained on many multilingual corpora, its performance may not be high for languages that are less used or have limited documentation in the training dataset.

3.2 DistilBERT

3.2.1 Overview

DistilBERT is an advanced natural language model (NLP), announced by Hugging Face in late 2019.

The goal: use distillation to reduce model size and increase processing speed while maintaining much of BERT performance.

3.2.2 DistilBERT vs BERT

DistilBERT is a small, fast, cheap and light Transformer model based on the BERT architecture. Knowledge distillation is performed during the pre-training phase to

- Reduce the size of a BERT model by 40%
- While retaining 97% of its language understanding abilities
- And being 60% faster.

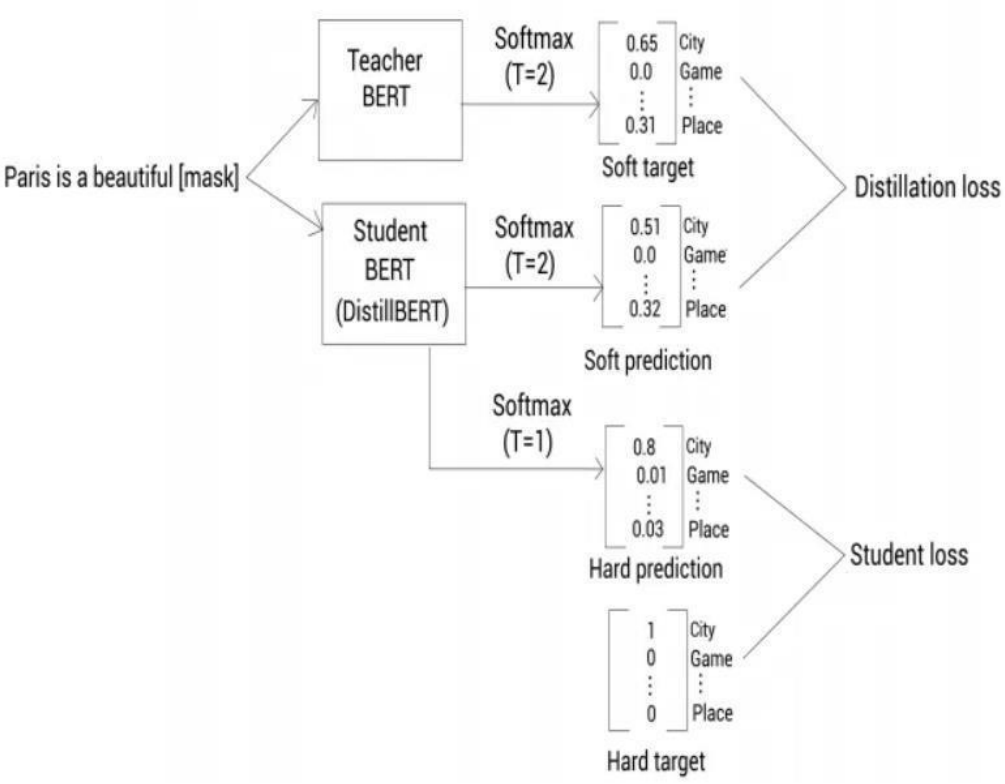
➔ ***DistilBERT Highlights:*** is an improved version of BERT. DistilBERT uses **knowledge distillation** to create a smaller, faster and more efficient version of the original BERT model while retaining much of its performance. With its compact size, DistilBERT is easy to deploy on resource-constrained devices that can be easily used for on-device applications.

➔ ***Disadvantages of DistilBERT:*** Reduced accuracy compared to BERT which may affect applications requiring absolute accuracy. Depends on the quality and performance of the BERT model as a teacher. If BERT has problems or limitations, these can be passed back to DistilBERT.

3.2.3 How does DistilBERT work?

Knowledge Distillation is a process in which a smaller model (student model) learns to reproduce the behavior of a larger, pre-trained model (teacher model). In the case of DistilBERT:

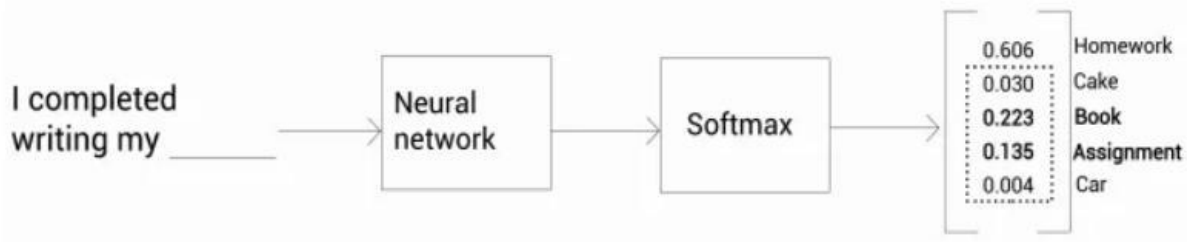
- Teacher model: The original BERT model, large and powerful.
- Student model: DistilBERT, a model learned from BERT to become a lighter and faster version.



Picture 3: DistilBERT simplyfied

3.2.4 What is knowledge Distillation?

The Teacher Network, Example below:

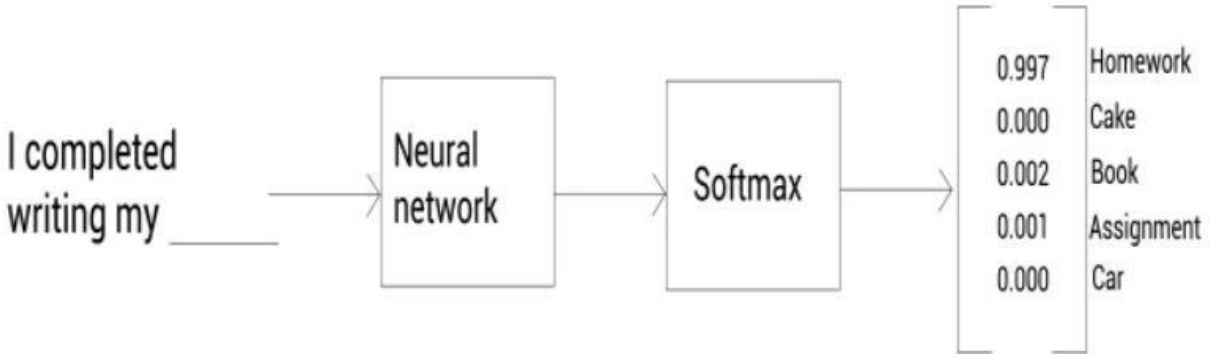


Picture 4: Example of teacher network

The Homework has the highest probability than others token,

Howerver the model can conclude other possibilities, this is call dark knowledge

During our knowledge transfer from teacher to student, we wish to learn this dark knowledge. But usually, we may train models that produce probabilities close to 1 for the best token like



Picture 5: Example of teacher network case when pribability distribution produced close to 1 or 0

Extracting dark knowledge is tough here as apart from ‘Homework’ because probabilities for all other tokens are ~0. So how do we extract dark knowledge in this case?

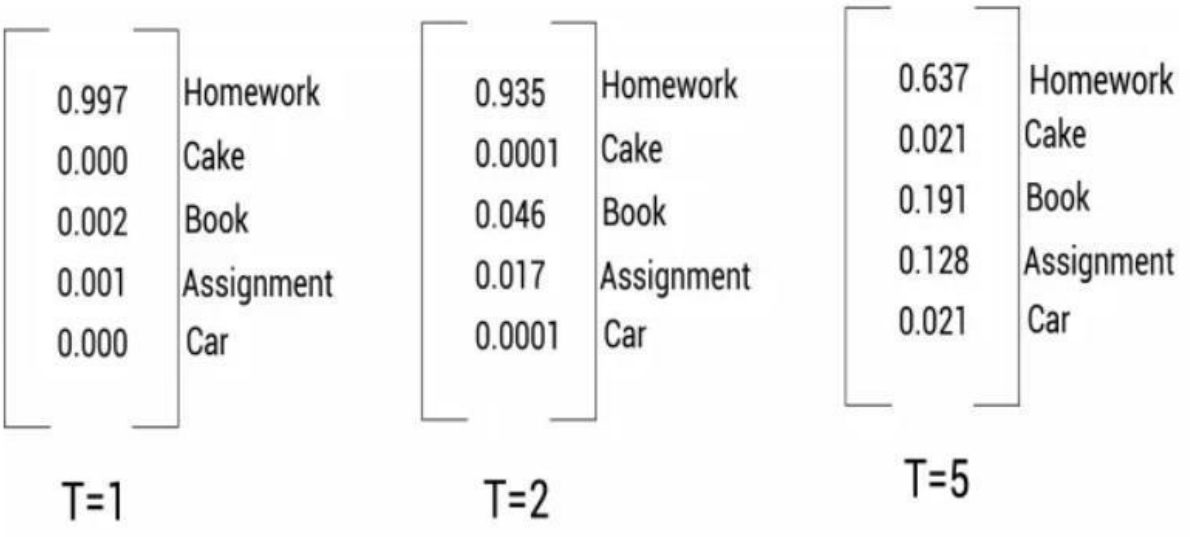
Using SoftMax Temperature

Replacing Softmax with SoftMax Temperature:

$$\frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \qquad \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$$

Picture 6: SoftMax Function (Left), SoftMax with Temperature (Right)

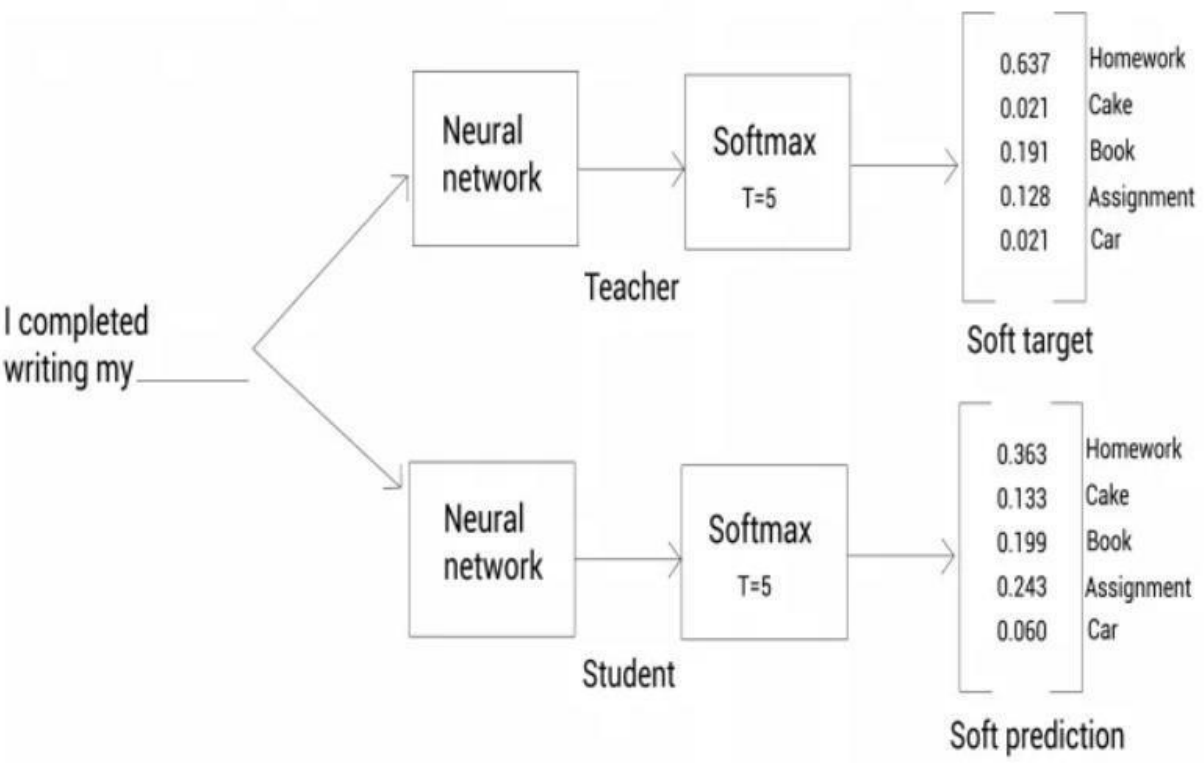
The constant T introduced is called Temperature helping in smoothing the probability distribution. A SoftMax function can be taken as SoftMax temperature with T=1. The bigger the value for T, the smoother the distribution. Let us see how probability distribution gets affected by different T values.



Picture 7: Hình 1: Softmax temperature with increasing T

The Student Network:

In the previous part, we have the pretrained Teacher Network with SoftMax Temperatute. Now is how do the Student Network learns from this.



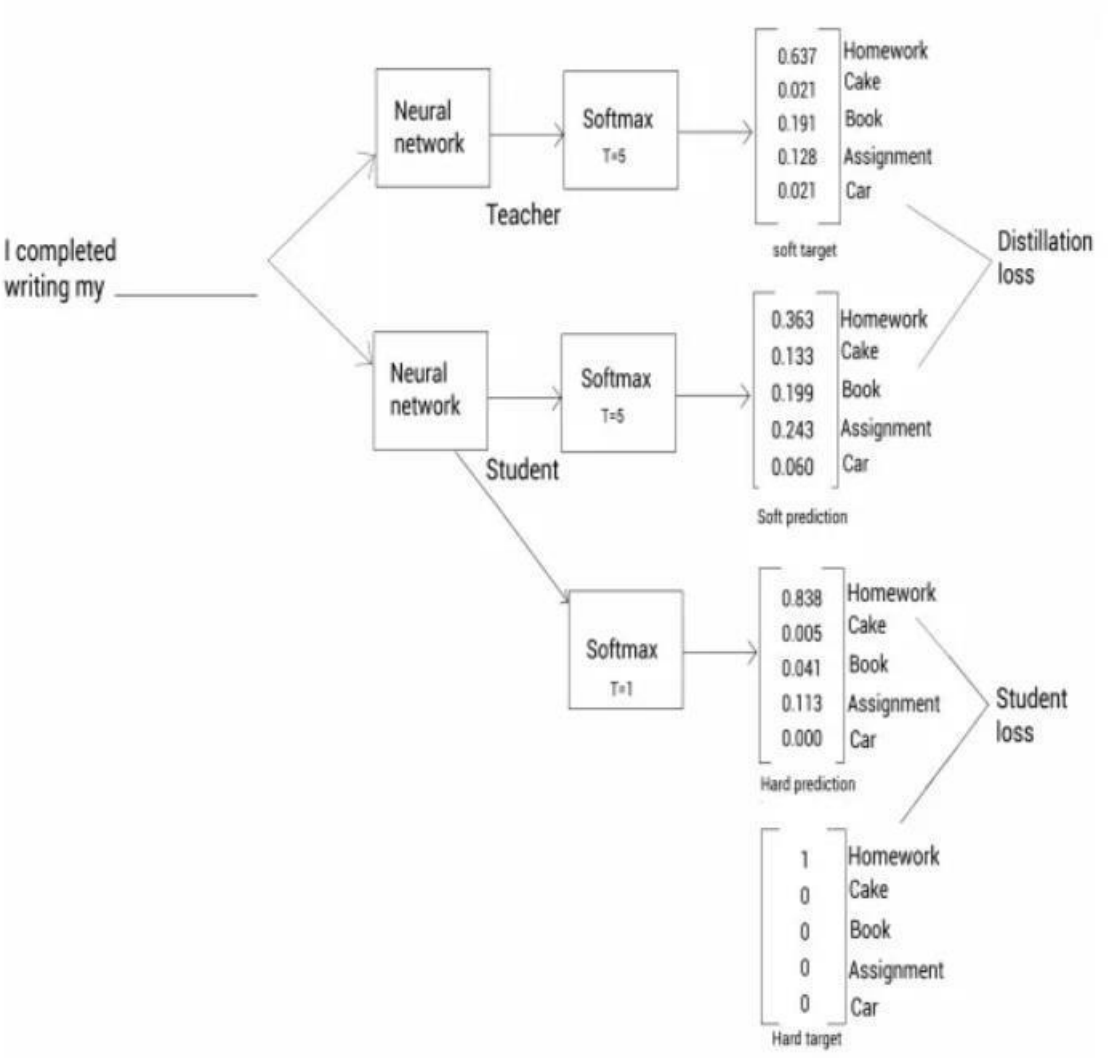
Picture 8: Teacher and Student Network

The probability distribution output by Teacher Network is called soft target.

The probability distribution output by Student Network is called soft prediction. <u>The pipeline of Student Network:</u>

The input sequence is fed to Teacher Network & 'Soft Target') is calculated.

The same Input sequence is fed to Student Network & a soft prediction is calculated.



Picture 9: Overall of Knowledge Distilation

The loss function has majorly 2 parts

Distillation loss: The cross-entropy loss between ‘Soft target’ & ‘Soft prediction’

Student loss: A couple more concepts are required

- Hard Target: Converting the ‘Soft target’ probability distribution into a one-hot-encoder sort of vector by setting the highest probability as 1 & the rest as 0

- Hard Prediction: In Hard Prediction, we keep T=1 & calculate the distribution of probability across all tokens
- Hence, in Distillation loss, T>1 but in Student loss, T=1.

3.2.5 Applications of DistilBERT

DistilBERT is used in various real-world applications due to its efficiency:

- *Chatbots and Virtual Assistants:* For faster response times and reduced computational costs.
- *Mobile and Edge Computing:* Deployment on devices with limited computational resources.
- *Web Services:* Improved speed and reduced operational costs in cloud-based services

Chapter 4. Demo code

4.1 Intsall library keras core and keras nlp

In this part, we download the keras core and keras nlp libraries, then set the backend. In this article, I will choose Tensorflow. In addition, we can also use the jax or torch backend.

```
!pip install keras-core --upgrade
!pip install -q keras-nlp --upgrade

import os
os.environ['KERAS_BACKEND'] = 'tensorflow'
```

Picture 10 Install library

4.2 Import library

Import the necessary libraries for the assignment

```
import numpy as np
import pandas as pd
import keras_core as keras
import tensorflow as tf
import keras_nlp
from sklearn.metrics import ConfusionMatrixDisplay, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
```

Picture 11: Import library

4.3 Load and process data

We read the given data including train.csv data for the machine to learn and test data for the machine to make predictions and give results.

```
df_train = pd.read_csv("/kaggle/input/nlp-getting-started/train.csv")
df_test = pd.read_csv("/kaggle/input/nlp-getting-started/test.csv")
```

Picture 12: Read datasets

4.4 Preprocess the data

Evaluate and filter necessary columns

```
BATCH_SIZE = 32
NUM_TRAINING_EXAMPLES = df_train.shape[0]
TRAIN_SPLIT = 0.8
VAL_SPLIT = 0.2
STEPS_PER_EPOCH = int(NUM_TRAINING_EXAMPLES)*TRAIN_SPLIT//BATCH_SIZE

EPOCHS = 1
AUTO = tf.data.experimental.AUTOTUNE
```

+ Code + Markdown

```
from sklearn.model_selection import train_test_split

X = df_train["text"]
y = df_train["target"]

X_train,X_val, y_train,y_val = train_test_split(X,y,test_size = VAL_SPLIT, random_state = 42)

X_test = df_test["text"]
```

Picture 13: Preprocessing

4.5 Load a DistilBERT model from Keras NLP

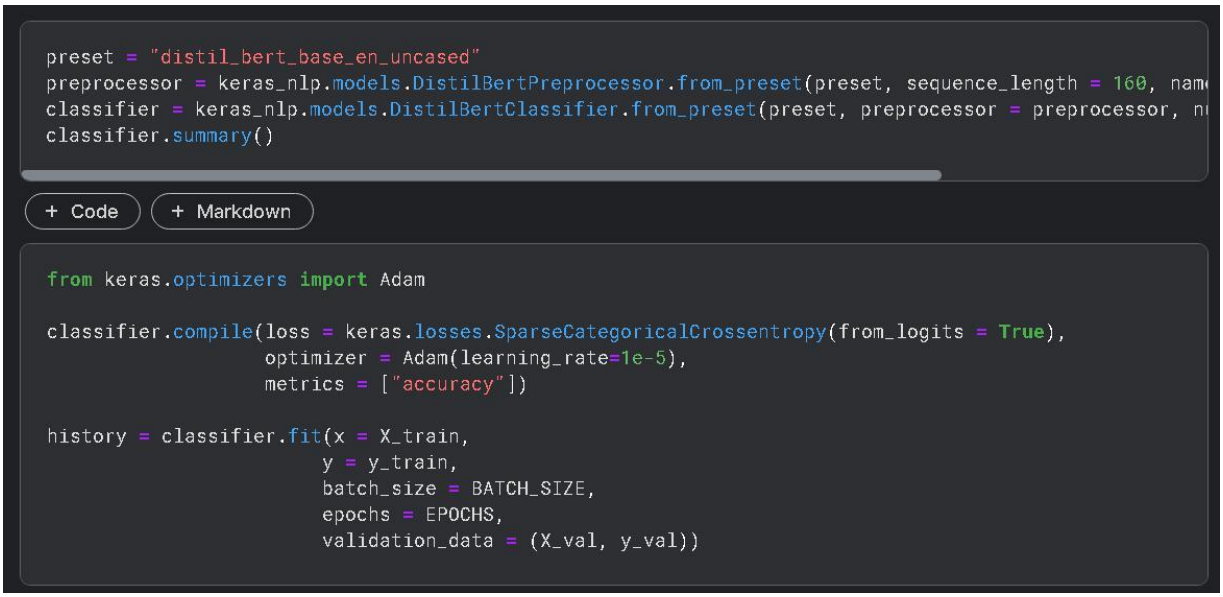
Load the BERT distillation model from Keras

```
preset = "distil_bert_base_en_uncased"
preprocessor = keras_nlp.models.DistilBertPreprocessor.from_preset(preset, sequence_length = 160, name = "preprocessor_4_t
weets")
classifier = keras_nlp.models.DistilBertClassifier.from_preset(preset, preprocessor = preprocessor, num_classes = 2)
classifier.summary()
```

Picture 14: Load DistilBERT

4.6 Train model

Mô hình chưng cấtBERT được lấy về từ kaggle và gọi để sử dụng sau đó ta tiền biên dịch và tiến hành huấn luyện mô hình vì mô hình huấn luyện khá lâu nên mình chỉ cho huấn luyện viên 1 epoch



```
preset = "distil_bert_base_en_uncased"
preprocessor = keras_nlp.models.DistilBertPreprocessor.from_preset(preset, sequence_length = 160, nam
classifier = keras_nlp.models.DistilBertClassifier.from_preset(preset, preprocessor = preprocessor, n
classifier.summary()
```

```
+ Code   + Markdown
```

```
from keras.optimizers import Adam

classifier.compile(loss = keras.losses.SparseCategoricalCrossentropy(from_logits = True),
                   optimizer = Adam(learning_rate=1e-5),
                   metrics = ["accuracy"])

history = classifier.fit(x = X_train,
                         y = y_train,
                         batch_size = BATCH_SIZE,
                         epochs = EPOCHS,
                         validation_data = (X_val, y_val))
```

Picture 15: Training model

4.7 Visualize and evaluate the result

The expression table above has the indexes from the cleanup matrix as follows:

❖ Not Disaster:

- True Negatives (TN): 3228 (Correct prediction “No disaster” is “No disaster”).
- False Positives (FP): 240 (Wrong prediction “No disaster” is “Disaster”).

❖ Disaster:

- False Negatives (FN): 667 (Wrong prediction of “Disaster” as “No Disaster”).
- True Positives (TP): 1955 (The correct prediction “Disaster” is “Disaster”).

From the above predictions, we can calculate the F1 score based on the following these formulas:

F1 is calculated as follows:

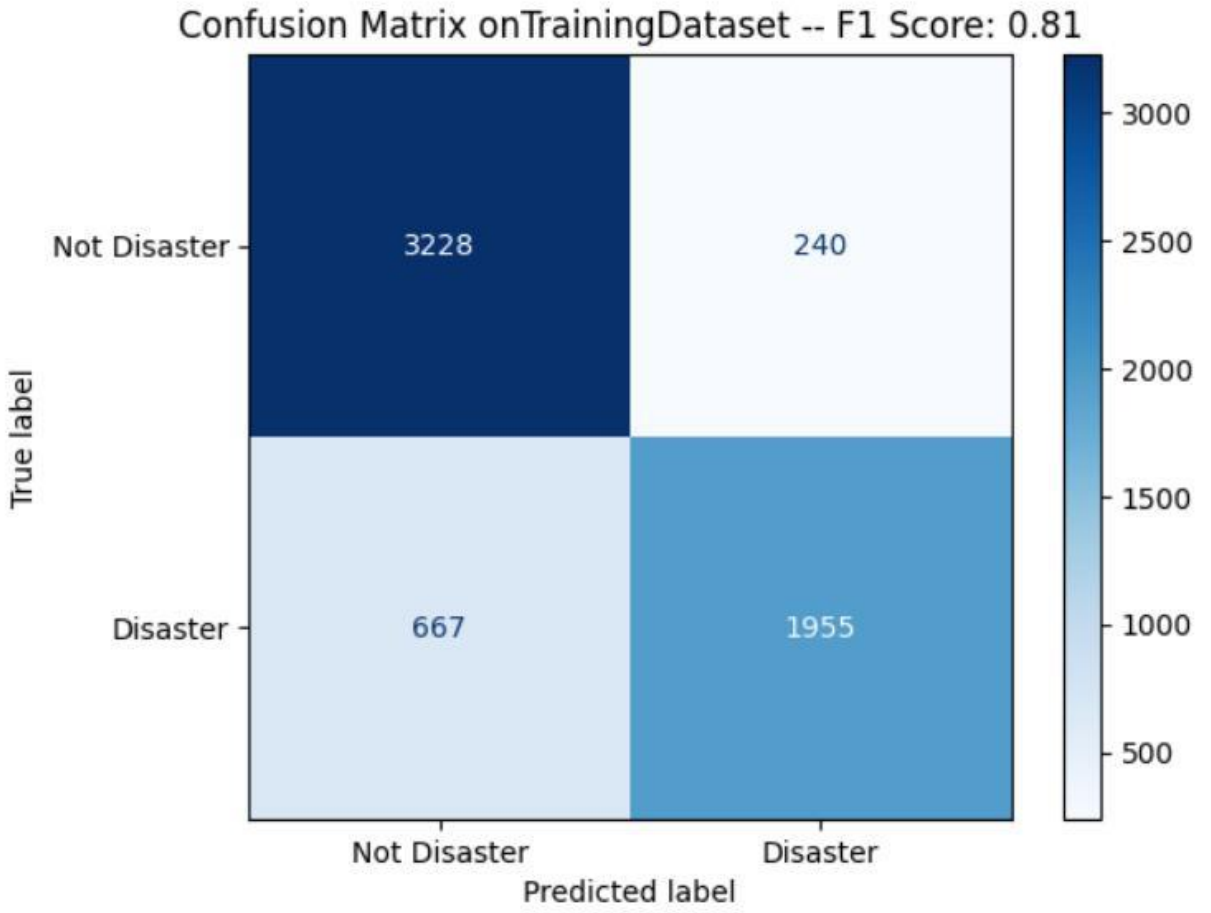
$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

where:

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

Picture 16: Fomulas of F1 score



Picture 17: Result on TrainingDataset

⇨ From the above predictions, we can calculate the F1 score based on the following two formulas

References

[1] d. P. C. Y. G. Addison Howard, "Natural Language Processing with Disaster Tweets," [Online]. Available: https://www.kaggle.com/competitions/nlp-getting-started/overview.

[2] M. G. f. G. I. S. P. C. Alexia Audevart, "KerasNLP starter notebook Disaster Tweets," [Online]. Available: https://www.kaggle.com/code/alexia/kerasnlp-starter-notebook-disaster-tweets#Generate-the-submission-file.

[3] Bischof Jonathan, "Getting Started with KerasNLP," [Online]. Available: https://keras.io/guides/keras_nlp/getting_started/.

[4] D. Sharma, "Introduction to DistilBERT in Student Model," [Online]. Available: https://www.analyticsvidhya.com/blog/2022/11/introduction-to-distilbert-in-student-model/.

[5] D. S. i. y. pocket, "DistilBert explained," [Online]. Available: https://www.youtube.com/watch?v=7PoLBeCjuSc.

[6] phamdinhkhanh, "Bài 36 - BERT model," [Online]. Available: https://phamdinhkhanh.github.io/2020/05/23/BERTModel.html.