BRIDGE i2i

INFORMATION → INSIGHT → IMPACT

Bridge-i2i's Automated Headline and Sentiment Generator

Inter IIT Tech Meet 9.0

- Problem Statement
- Overall Approach
- Advantages of Transformers
- Task 1: Sentiment Classification
- Task 2: Aspect Based Sentiment Classification
- Task 3: Headline Generation
- Suggestive improvements for real-time deployment.

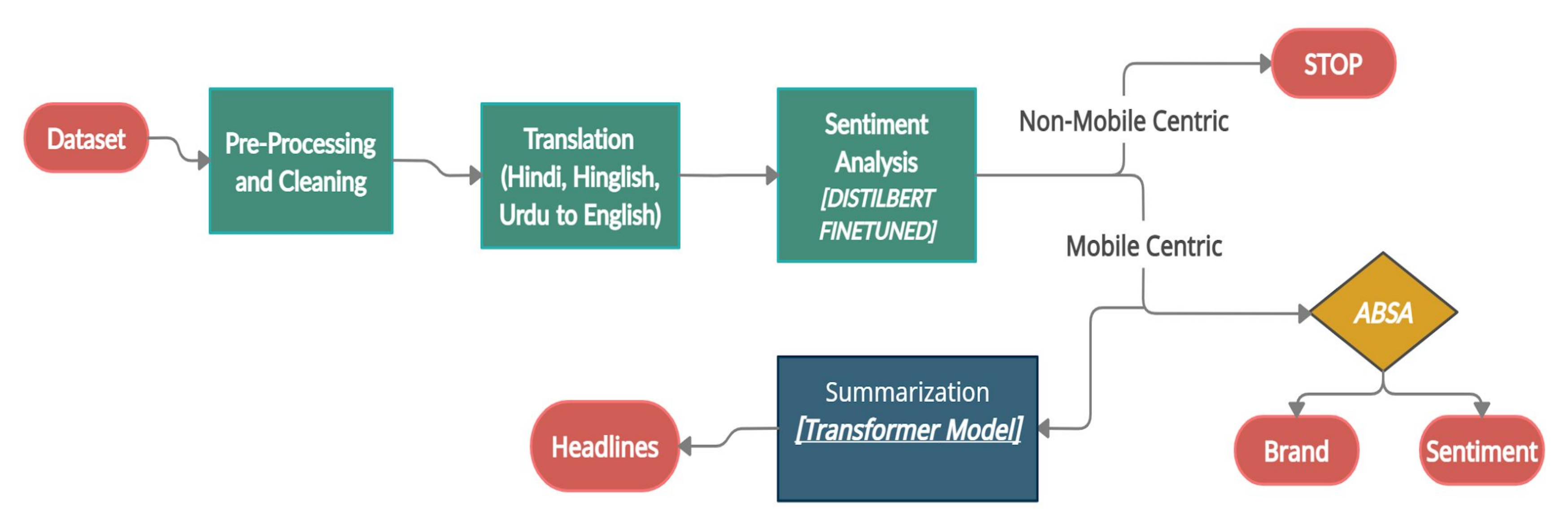
Presented by:
Team **14 (H2-B2I-14)**

Problem Statement

BRIDGE i2i
INFORMATION - INSIGHT - IMPACT

- ❖ Develop an intelligent system that could first identify the theme of tweet and articles.
- ❖ If the theme is related to mobile technology then it should identify the sentiments against a brand.
- ❖ A one sentence headline of 20 words for articles having mobile centric articles should be generated.

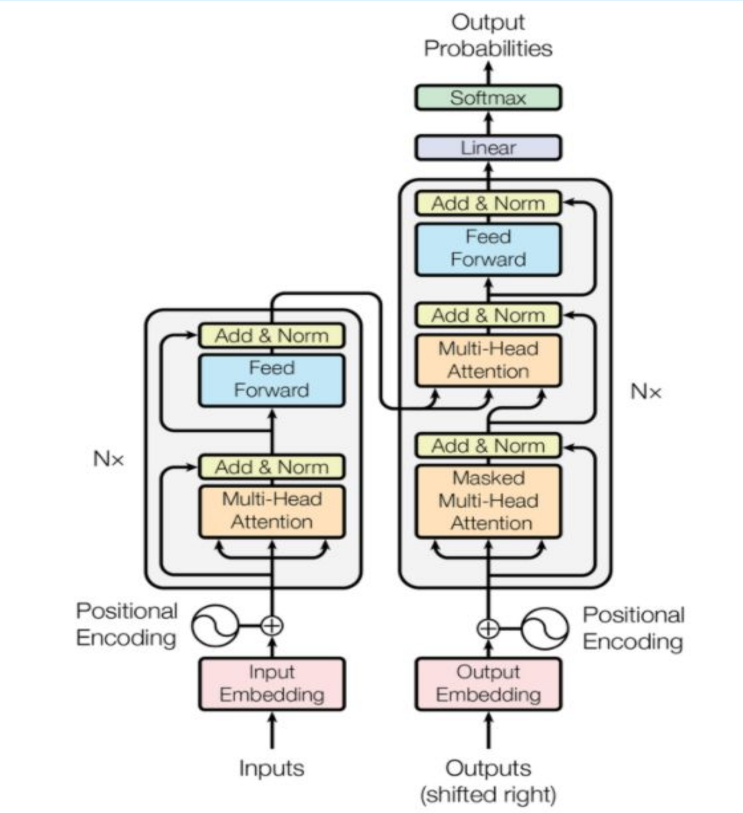
Approach



Dataset
Pre-Processing and Cleaning
Translation (Hindi, Hinglish, Urdu to English)
Sentiment Analysis [DISTILBERT FINETUNED]
Non-Mobile Centric
STOP
Mobile Centric
ABSA
Brand
Sentiment
Summarization [Transformer Model]
Headlines

Why prefer Transformers?

Transformers are better than all the other architectures because they totally avoid recursion, by processing sentences as a whole and by learning relationships between words thanks to multi-head attention mechanisms and positional embeddings. Unlike RNNs, Transformers do not require that the sequential data be processed in order.



(Flow Diagram of a typical Transformer Architecture)

PREPROCESSING

Preprocessing

ARTICLES

The following were removed from the article dataset while training

- ❖ HTML Tags
- ❖ URL links
- ❖ Irrelevant information such as menu texts and author's information.
- ❖ Noisy articles were also discarded.

TWEETS

The following were removed from the tweet dataset while training

- ❖ Twitter specific lexicals such as retweets and mentions
- ❖ Reserved words (RT and FAV)
- ❖ URL links
- ❖ Empty placeholders

Translation

I. Translation of the text with languages different from english was done using an API utilizing free version of Google Translate.

II. Google Translate supports 100+ languages, hence taking care of multilingual texts in the dataset. More so, Indic languages such as Bengali, Punjabi, Tamil, Urdu and Telugu are also supported, which allows it to be easily scaled to articles and texts belonging to a wide range of languages.

TASK 1

SENTIMENT CLASSIFICATION

Sentiment Classification

OBJECTIVE:

To classify the theme of tweets/articles; whether they are mobile-centric or not.

- **DistillBERT** model was used for this task.
- **DistillBERT** is an efficient, fast and lightweight Transformer model trained by distilling the traditional *bert-base-uncased* model.
- It contains 40% less parameters than *bert-base-uncased* model.
- Owing to the above fact, it runs 60% faster than former while preserving 95% of BERT's performance as measured on the GLUE language understanding benchmark.

Comparison of different BERT models

BRIDGE i2i
INFORMATION - INSIGHT - IMPACT

	BERT	RoBERTa	DistilBERT	XLNet
Size (millions)	**Base**: 110 **Large**: 340	**Base**: 110 **Large**: 340	**Base**: 66	**Base**: ~110 **Large**: ~340
Training Time	**Base**: 8 x V100 x 12 days* **Large:** 64 TPU Chips x 4 days (or 280 x V100 x 1 days*)	**Large:** 1024 x V100 x 1 day; 4-5 times more than BERT.	**Base**: 8 x V100 x 3.5 days; 4 times less than BERT.	**Large:** 512 TPU Chips x 2.5 days; 5 times more than BERT.
Performance	Outperforms state-of-the-art in Oct 2018	2-20% improvement over BERT	3% degradation from BERT	2-15% improvement over BERT
Data	16 GB BERT data (Books Corpus + Wikipedia). 3.3 Billion words.	160 GB (16 GB BERT data + 144 GB additional)	16 GB BERT data. 3.3 Billion words.	**Base**: 16 GB BERT data **Large**: 113 GB (16 GB BERT data + 97 GB additional). 33 Billion words.
Method	BERT (Bidirectional Transformer with MLM and NSP)	BERT without NSP**	BERT Distillation	Bidirectional Transformer with Permutation based modeling

Fine Tuning

- Fine-tune training was done with a 80-20 train and validation split of the entire dataset for 10 epochs, using Keras framework.
- Training early-stopped at 6th epoch, with average time per epoch = 2.35 minutes.
- The training accuracy stood at 99.5%, while validation accuracy came out to be 96.57%.
- We obtained an F1 Score of 0.9725 with the recall as 0.9941 over the training dataset.
- Because of very small training time, the approach is scalable and suited for edge computing based applications.

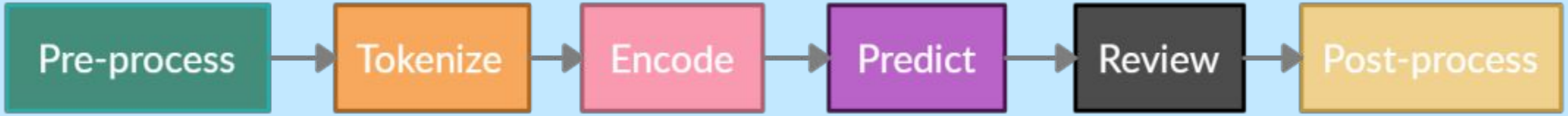
TASK 2

ASPECT BASED SENTIMENT ANALYSIS

Aspect Based Sentiment Analysis (ABSA):

- ❖ It is a text analytic technique that categorizes the data according to a predefined aspect and then investigates multiple sentiments associated with different objects present in the text.
- ❖ Implemented using python package *aspect-based-sentiment-analysis.* It utilizes the Ada-BERT model.
- ❖ Ada-BERT leverages differentiable Neural Architecture Search to automatically compress BERT into task adaptive small models for specific tasks.
- ❖ Hence, can be deployed for practical applications.

Aspect Based Sentiment Analysis (ABSA):



(aspect-based-sentiment-analysis Library Workflow)

❖ It has an internal component called Professor, which reviews and supervises model behaviour. This ensures monitoring of the model when it is served.

❖ The Preprocessing Stage takes in aspect entity and input text, and converts them into a pairs of text and an aspect, for each aspect entity.

- The BERT-ADA model used in trained in two stages- a self supervised training on domain specific corpus, followed by fine-tuning on the said task.

- The model incorporates specific attention to tokens related to the given aspect (constraint-based), and accordingly predicts the sentiment of the aspect.

- If a brand name was located, we searched for a window of characters from that brand's name to find out the sentiment associated with that company.

- The process took a mere runtime of 4.25 minutes to complete the task.

- Hence, considering the small iteration time and the proficiency of ABSA to extract granular information, the approach is scalable and efficient.

TASK 3

HEADLINE GENERATOR

Headline Generation

OBJECTIVE: To generate headlines for articles classified as mobile-centric.

The Headline Generation can be thought of as a Summarization based task, with the output sequence length below a certain limit.

We tested a total of four architectures for our purpose, they are:

- Pegasus
- T5
- Pegasus with BART
- T5 with BART

Basic Outline

- ❖ The T5 and Pegasus models were fine-tuned on the given corpora using Torch libraries.
- ❖ The decoder length was modified to better reflect the nature of the task.
- ❖ The training times for the two models are quite less; both of the models were trained for a total of 10 epochs, out of which:
  - ➢ T5 had an average time of 11 minutes/epoch
  - ➢ Pegasus had an average time of 17 minutes/epoch
- ❖ The inference time per example translated to a mere 4.53 seconds.

Advantages of this approach

- Relatively low Inference and Training times imply that this pipeline is both scalable and efficient in nature.
- The encoder and decoder length can be explicitly specified during the fine-tuning process to better adapt to the dataset and the nature of the task.
- The entire pipeline is based upon the Transformer architecture, which is superior for tasks involving Conditional Generation such as Summarization.
- Basic preprocessing such as removing punctuations, stopwords and other tasks are taken care of by the pre-trained model-specific tokenizer.

SUGGESTIVE IMPROVEMENTS FOR REAL-TIME DEPLOYMENT

Suggestions

The solution proposed for this problem statement can be effectively scaled to support real-time applications, including edge-computing scenarios. However, we hereby list a number of suggestions which can be implemented to further improve upon the offered workflow:

- Using **Longformer** transformer model to process articles having a large length, and capture better long range dependencies for the same.
- Using a paid Google Translate API for translation of multilingual data to English for further processing.
- Fine-Tuning AdaBERT on a custom dataset to improve the performance of the Aspect Based Sentiment Classification task for a given application.

Thank You!