Team name: Natural Language Processors
Naman Khurpia - namankhu@buffalo.edu, 50478714
Mamatha Yarramaneni - mamathay@buffalo.edu, 50466676
Mohammed Junaid Shaik - mshaik6@buffalo.edu, 50468349

INTRODUCTION - This project aims at classifying and generating spoilers for clickbait posts collected from social media posts by using natural language processing techniques.

PROBLEM DESCRIPTION – Clickbait posts are posted links and advertisements that generate curiosity instead of giving a proper description of what the article is about. A model that classifies and generates a small text that satisfies the curiosity by summarising the gist behind the post is proposed. The problem statement consists of 2 tasks, Task 1 is to classify the spoiler type as a phrase, passage, or multi. Task 2 is detecting spoilers from the linked document.

The problem of spoiler classification has been implemented as a text classification task. The problem of spoiler generation has been implemented as a passage retrieval and question-answering system, and we believe to narrow it down to the core problem we can simply summarise the content and then try it as a question-answering system.

Task 1 approaches - **Naïve Bayes, Logistic classification, BIRNN Network, BERT Model**
Task 2 approaches - **IR-based Model, LSTM-based Model, DistilBERT Model**

TASK 1 ARCHITECTURES

1.1 Naive Bayes, 1.2 SVM Classification, 1.3 Logistic Regression

Figure 1

1.4 BiRNN Network

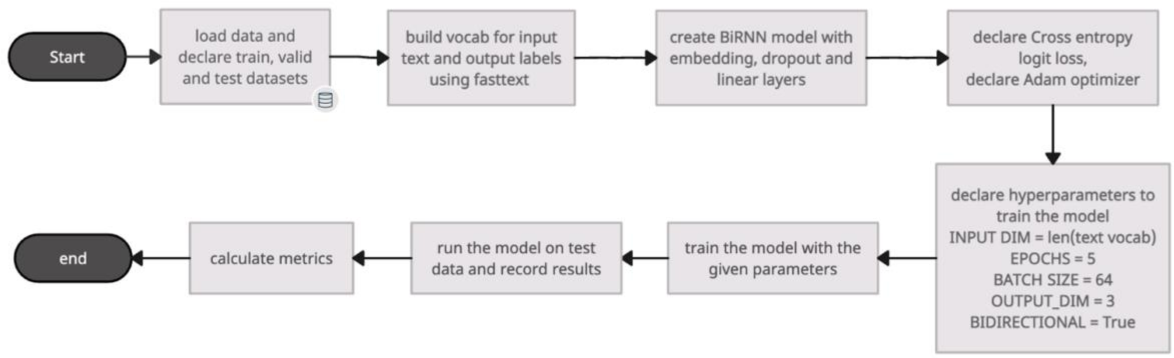

Figure 2

1.5 BERT Model (Baseline)

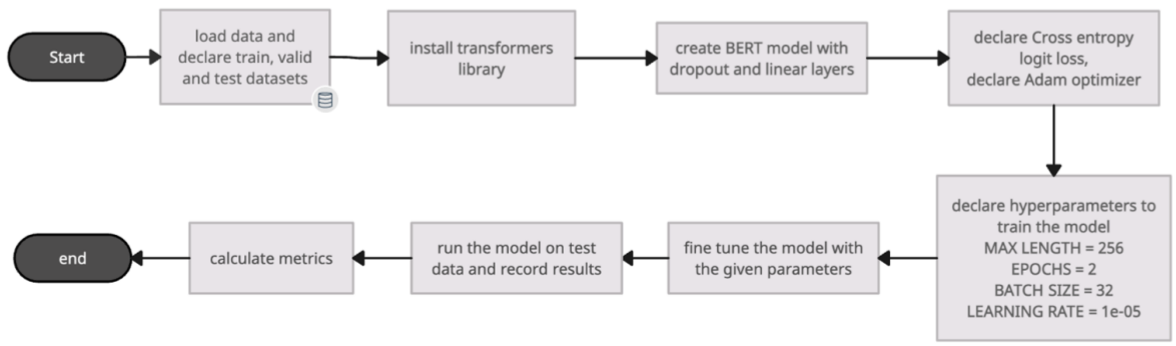

Figure 3

TASK 1 RESULTS (test results for 800 posts: phrase-335, passage-322, multi-143)

Model	Accuracy	Macro-averaged F1 score
Naive Bayes	0.57375	0.46440793063022595
SVM Classification	0.54375	0.522308796726875
Logistic	0.54375	0.5200399321154038
BiRNN	0.5115	0.49457
BERT base cased	0.63125	0.624228179465244

TASK 1 ANALYSIS AND AREAS OF IMPROVEMENT

The models were trained and validated on 3200 posts and tested on 800 posts. Among the classic models, Naive Bayes performs better than other models in terms of accuracy of 57%. The input to the neural models is post-concatenated with the linked document. The BiRNN model could not retain the longer context and performed poorly. The BERT transformer model has self-attention which retained the context of the input text for longer sentences and outperformed the other models.

The higher versions of BERT transformer models like DeBERTa and RoBERTa can be used for further improvement in the accuracy of the prediction.

TASK 2 ARCHITECTURES

2.1 IR-Based Model (created from scratch)

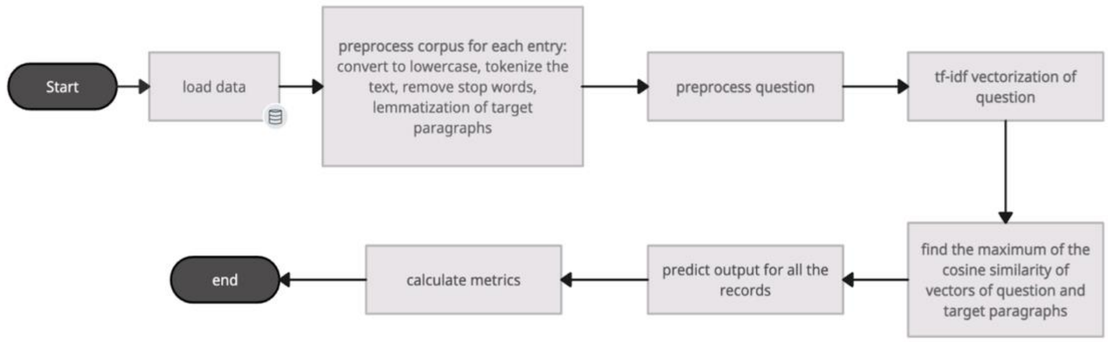

Figure 4

Baseline results:(test results for 800 posts: phrase-335, passage-322, multi-143)

Type	BERT precision score	BERT recall score	BERT F1 score	BLEU score	METEOR score
phrase	0.7987	0.8402	0.8183	0.0045	0.1552
passage	0.8525	0.8759	0.8636	0.1084	0.3010
multi	0.8398	0.8320	0.8353	0.0291	0.1666

2.2 LSTM-Based Model (created from scratch)

Baseline Architecture - We first preprocessed the text data by tokenizing the passages into sequences of words or characters and converting them into numerical vectors that can be input into the LSTM. Once we have prepared the data, we train the LSTM model using a binary classification objective, with the goal of predicting whether a given passage contains a spoiler and identifying it. Figure 5 shows the pipeline and architecture diagram for the LSTM model.

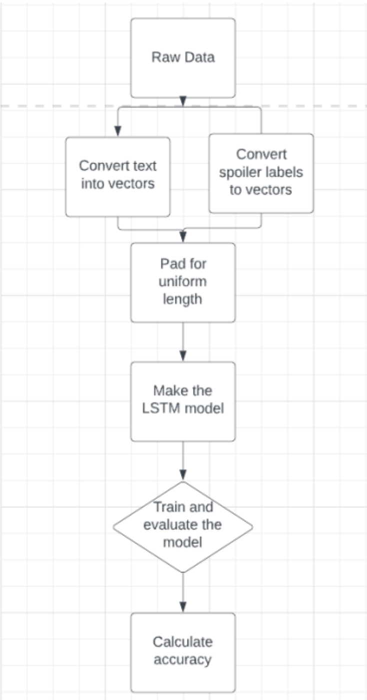

Figure 5

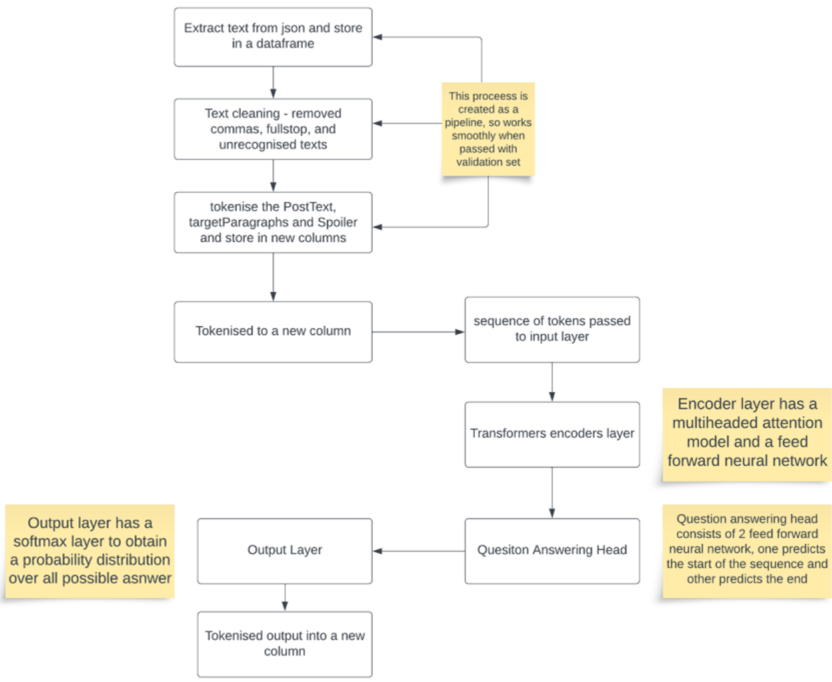

Figure 6 DistilBERT Model (LLM) architecture and pipeline for Training and Validation Set - using Transformers Library

Baseline results:(test results for 800 posts: phrase-335, passage-322, multi-143)

Type	BERT precision score	BERT recall score	BERT F1 score	BLEU score	METEOR score
phrase	0.1327	0.1331	0.1328	0.1693	0.0921
passage	0.1473	0.1510	0.1491	0.0663	0.0472
multi	0.0624	0.0792	0.0698	0.0082	0.0093

Analysis: We observe that the scores obtained are very low and not acceptable. Mostly this is happening because the LSTM model is generating its own sentences or adding new words which are not there in the passage. Though on manual checking it feels some more of the answers are right, because of the addition or modification of the sentences, they no longer match the ground truth. After the very low scores, we abandoned this approach and moved on to an LLM-based model, so that pre-trained models can better understand text and context and give better replies.

2.3 DistilBERT Model (LLM - using transformers library)

Baseline Architecture - DistilBERT is a distilled version of the pre-trained BERT model, which is fine-tuned of SQuAD (Stanford Question Answering Dataset), so it generates answers based on the given context. Figure 6 explains the pipeline and the architecture of the model very clearly and the results are given below-

Baseline Results: (test results for 800 posts: phrase-335, passage-322, multi-143)

Type	BERT precision score	BERT recall score	BERT F1 score	BLEU score	METEOR score
Phrase	0.8717	0.8708	0.8704	0.3277	0.2225
Passage	0.8518	0.8273	0.8390	0.1209	0.0868
Multi	0.8415	0.8060	0.8230	0.0929	0.0607

TASK 2 ANALYSIS: - The scores like BLEU score, BERT score, and METEOR scores are far from perfect, the reason being they check for n-gram matching. The predicted results are compared to the ground truth. The golden spoiler cannot be detected until our model is trained enough to understand the post text. Our approach of passage retrieval works well in the case of passages. The main problem in phrase seems to be, finding the question in the postText field, the golden spoiler cannot be detected until our model is trained enough to understand the postText, in most cases the postText doesn't start with a What, Why, Which, and therefore we get poor performance out of the model but if we generate a proper question from the postText we can find the golden spoiler hidden in target paragraph.

TASK 2 AREAS OF IMPROVEMENT: -

- For the multi-clickbait type, we plan to all retrieve passages that are relevant to the given clickbait post and then retrieve the answer from each of the passages and concatenate.
- Increasing the number of heads for the multi-headed attention model for spoiler extraction can help us improve the results.
- We plan to use a better Question Answer model with a rephrased postText that better captures the essence of the clickbait post.
- Using advanced transformers models for much better results.

```
[ ] question = "Say it ain't so! Jon Stewart has set his official departure date from #TheDailyShow"
    context =  "Jon Stewart now has a firm departure date from Comedy Central's \The Daily Show.\ The comic announced on Monday's broadcast of the program that he will leave the sho

    response = qa_pipeline(context=context, question=question)

    print(response["answer"])

    Comedy Central's \The Daily Show

[ ] question = "what is the date?"
    context =  "Jon Stewart now has a firm departure date from Comedy Central's \The Daily Show.\ The comic announced on Monday's broadcast of the program that he will leave the sho

    response = qa_pipeline(context=context, question=question)

    print(response["answer"])

    August 6th
```