Is that headline Clickbait?

Charmi Chokshi
Student (261130926)
School of Computer Science,
McGill University
charmi.chokshi@mail.mcgill.ca

Sandeep Kumar
Student (260968812)
School of Computer Science,
McGill University
sandeep.kumar@mail.mcgill.ca

Xing Han Lu
Mentor
School of Computer Science,
McGill University
xing-han.lu@mila.quebec

Abstract

To catch the reader's attention digital and print media are using "Clickbait" headlines. To their monetary benefits, they are misleading the population by publishing catchy headlines to get more user engagements and clicks per post. Nowadays, it has grown to become a nuisance to social media users and operators. Malicious content publishers misuse social media to manipulate users and exploit celebrities or well-known personalities many times. There are various ongoing research on better clickbait detectors Kaur et al. (2020) and Anand et al. (2017) that try to classify if a given headline is clickbait or not. In this project, we built classical ML-based approaches and used recent transformer-based architectures to find an answer to a fundamental question if only a new article's headline is sufficient to successfully classify if it is clickbait, or would we also need more context from the related body of the article? We came to the conclusion that just a headline is enough to build a decent classifier which even requires low computing. We fine-tuned DEBERTA-base He et al. (2020) on Webis-17-dataset which lead to 87.51% test accuracy and an F1 score of 0.92. We also proposed a novel method to firstly, generate new headlines from the article by fine-tuning T5-base Raffel et al. (2020) and secondly, finding similarities between the original and generated headline's embeddings and using a threshold for the classification task.

1 Introduction

News headlines, be it print media or digital media, have long been criticized for being "Clickbait". A headline is Clickbaity when it is, arousing curiosity instead of providing informative summaries to its reader. In addition to news articles, with the increase in online marketing and CPC-like (Cost Per Click) money-making models, various agencies are generating catchy titles for their posts so as to get more user clicks and engagements. Humans have a curious nature by default and the main reason that results in clicking on intriguing posts is due to the "Curiosity Gap" the headline creates. Loewenstein's information-gap theory of curiosity Loewenstein (1994) is usually presented as the psychological reasoning behind this. Nowadays the problem of clickbait has increased to such an extent that people don't mind fake and unrelated headlines to articles in order to generate ad revenues from the posts. Typically, it is spread on social media in the form of short teaser messages that may read like the following examples:

1. Drink this formula every morning to lose 5kg of weight in 5 days
2. Giant asteroid hurtling towards Earth!
3. 9 Out Of 10 Americans Are Completely Wrong About This Mind-Blowing Fact
4. Here's What Actually Reduces Gun Violence

When reading such messages, readers get the distinct impression that something is odd about them, some emotional reaction is promised, something unnamed is referred to, and some lack of knowledge is ascribed. From a user's standpoint, clicking on these catchy titles and reading the following paragraphs which majorly have no connection with the headline is a waste of time and is misleading. As a step towards stopping this social media nuisance and promoting responsible journalism as well as to stop the spread of misleading information in today's information world, we propose the project on identifying if a given headline is a clickbait or not by using Natural Language Processing (NLP) techniques. It can have a wider use-case as an add-on extension on web browsers showing if the headline is clickbait and the user should click on it or not.

Well-known media and ad platform, Facebook for example have found that people don't like stories that are misleading, sensational or spammy.

That includes clickbait headlines that are designed to get attention and lure visitors into clicking on a link. They even launched a war Babu et al. (2017) against clickbaits in 2017 to utilize users feedback to identify and block domains from their platform forever that are notorious for producing clickbaits.

Previous work on clickbait detection includes Potthast et al. (2016a) (specific to the Twitter domain) and Chakraborty et al. (2016) which rely on a rich set of hand-crafted features by utilizing existing NLP toolkits and language specific lexicons. But it is often challenging to adapt them to multilingual or non-English settings since they require extensive linguistic knowledge for feature engineering. Anand et al. (2017) have introduced a neural network architecture based on Recurrent Neural Networks for detecting clickbaits and Kaur et al. (2020) proposed CNN-LSTM model. But they only used character and word-level embeddings where as we believe sentence-level embeddings would outperform them and hence, we used Sentence-Transformer Reimers and Gurevych (2019) to get sentence-level embeddings.

Not just treating this as a classification task, we found an answer to a fundamental question of whether only a new article's headline is sufficient to successfully classify if it is clickbait or if would we also need more context from the related body of the article. We used the Webis Clickbait Corpus 2017 which has 40,976 samples of headlines, passages, keywords, captions, etc. from major US news publishers. We implemented 8 different models starting with using just Headline as a feature on SVM, and XGBoost as well as on transformer-based models ELECTRA Clark et al. (2020), and DEBERTA. Furthermore, we fine-tuned ELECTRA and DEBERTA with Headline and Article features and lastly we performed a text summarization task on the Article using the T5-base of-the-shelf model and fine-tuned model. We have performed ablation studies to understand how well the newly generated headlines are and have tried various different thresholding techniques to achieve the best classification result.

2 Related Work

In the past there have been multiple approaches to solve the problem of clickbait detection. A wide range of methods from classical ML and deep learning sequence models have been explored in this field. New advances like transformers can also present a way forward in this task.

Classical ML Approaches: Initially the researchers were utilizing classical machine learning techniques and heavily utilized feature engineering techniques to address the problems. In Click Bait Detection by Potthast et al. (2016b) the authors creates multiple types of features like word n-grams, and char n-grams and then perform Chi-square tests to select the best features which are used in classifier models such as Random Forests, Logistic Regression, and Naive Bayes. They found that using Random Forest with selected features yields the best ROC-AUC metric. In Chakraborty et al. (2016) authors utilize different ML models specifically SVMs to achieve high accuracy in clickbait classification. These approaches were great solutions at the time the competition was launched. But, nowadays there are many deep learning-based approaches that outperform these by a huge margin in many NLP tasks. In the next section, we will discuss more of those.

Deep Learning Based Methods: One of the early strides in the application of deep learning in the domain of clickbait analysis was successfully applied by Agrawal (2016) where CNN was utilized for the task. Two models, one from scratch and one using Word2Vec Embeddings trained by Mikolov et al. (2013) and were able to achieve great results. In 2017 the ClickBait Challenge Potthast et al. (2018) was launched in order to tackle the problem of Clickbait articles on Twitter. This also saw some pretty amazing results from the deep learning area. Anand et al. (2017) used sequence-based models along with distributed word embedding and character level word embedding and found BiLSTMs to work really well on the challenge (Bi-GRUs a bit worse with 1% less accuracy). Zhou (2017) then used token level self-attentive networks, particularly Bi-GRUs on the Webis Dataset and were able to achieve 85.6% accuracy on it.

After the arrival of attention mechanism, Bahdanau et al. (2014) the field of NLP advanced a lot. Attention paved the way for the Transformer architecture by Vaswani et al. (2017) which completely revolutionized the domain and more and more people switched to large pre-trained model rather than using LSTM and GRU-based sequence models. One such architecture which took things to the next level was BERT or Bidirectional Encoder Representations from Transformers by Devlin et al. (2018). It is based on transformer architecture and trained bidirectionally using the encoder module (as shown in 1). Trained by utilizing Masked Language modelling and Next sentence prediction tasks it was able to outperform all other models on most NLP tasks. Transformer models like BERT can be fine-tuned for downstream tasks and be applied to a different problem set. Clickbait classification can be one such downstream task where the model can be fine-tuned to its specific data.

Since the discovery of transformer-based architectures, the Webis Clickbait Detection Challenge did also see these new methods being applied to it. In Indurthi et al. (2020) the authors were first to approach the regression aspect of the Webis Clickbait challenge using transformers and measured the clickbait strength to identify clickbaits. They used various regression-based models on top of the transformer representations to generate clickbaitness score to classify the data. With the fine-tuned variants they were able to push the accuracy levels to 85.7%. Then in Rajapaksha et al. (2021) the authors use pooled outputs of different layers of models like BERT, XLNet, etc along with different heads (linear, RNN based, and non-linear) to study the behavior on Clickbait Classification Task. They used Kaggle Clickbait detection task (Train a clickbait detector dataset) as a test set and achieve about 85.93% accuracy using a combination of pooled output from RoBERTa architecture and a linear layer on top.

Apart from the above-mentioned ones, one another approach to the clickbait challenge we could utilize is the article summaries. Transformers with sequence-to-sequence architecture have also made strides in this field of Document Summarization. Models like Bart, Lewis et al. (2019) and T5, Raffel et al. (2020) were able to achieve state-of-the-art performance in summarization tasks. These pretrained models are readily available and can be utilized in summarizing the articles which we hope to use in this project. Once we have the summaries of articles, we can find the embedding of that text summary and the distance of it with the embedding of the actual news headline. Various distance metrics or Cosine Similarity can be used to understand if the summary matches the headline or not, if not that is Clickbait.

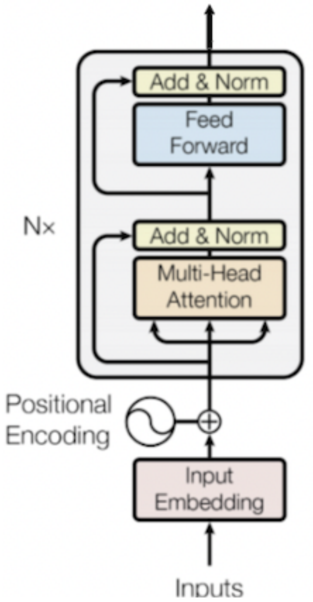


Figure 1: Encoder block of Transformer (Vaswani et al., 2017)

3 Modelling

We implemented 8 different models in total namely SVM, XGBoost, fine-tuned ELECTRA, and DEBERTA on the headline and headline as well as article text. Furthermore, we performed a text summarization task on the Article using the T5-base of-the-shelf(trained for headline generation) model and fine-tuned model. We have performed ablation studies to understand how well the newly generated headlines are and have tried various different thresholding techniques to achieve the best classification result. Below is in detail about each of these methods.

All the experiments were ran on a single NVIDIA Tesla V100 16GB GPU on the Google Cloud Platform.

3.1 Baselines

As this a binary classification problem with not an immense amount of text data, we thought of trying out simpler classifiers (non-neural network) models to create baselines. We have used Support Vector Machine (SVM) and Extreme Gradient Boosting (XGBoost) classifiers for the clickbait classification task. The objective of SVM algorithm is to find a hyperplane in an N-dimensional space that distinctly classifies the data points. We used RBF (Radial Basis Function) kernel for this computation. XGBoost is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library. Our setup for the XGBoost had 150 estimators and the maximum depth allowed for each tree was 3 after trying out different values. Processing on input text such as removing special characters and emojis, substituting website links, and removing Null values and unwanted columns has been done. As a result, SVM achieves nearly 81% accuracy and XGBoost about 79% thus, SVM performs slightly better than the XGBoost.

3.2 Proposed Models

We performed fine-tuning of transformer-based models on this task and also, proposed a novel method for classification by doing a text summarization/headline generation.

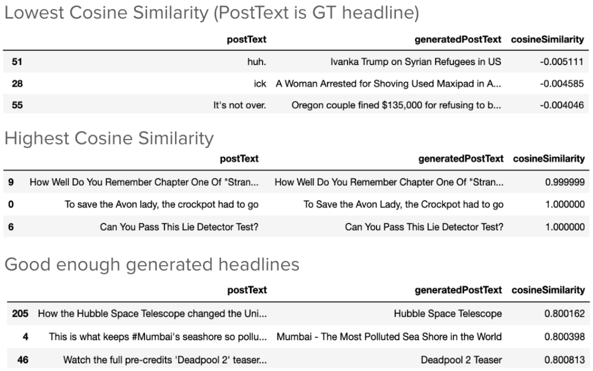
3.2.1 Fine-tuning Transformers

To see if only using the headline's text is sufficient for this task of not we fine-tuned two transformer models on the headline and on headline and article. The choice of models were Electra-base and Deberta-base as they are considered to be state-of-the-art for classification task and are also lightweight. For headline we use 'PostText' and for Article we combine 'TargetParagraphs' from the data. Processing on input text such as removing special characters, removing Null values and unwanted columns has been done. Firstly, we fine-tuned both the models with headline feature's embedding received from respective tokenizers. Both the models were trained for 3 Epochs, with 2e-5 Learning Rate, 0.3 Dropout, 0.05 Weight Decay, and with 128 Max Sequence Length. The batch size was chosen based on the memory availability on the GPU as Deberta-base is a heavier model than Electra-base we used a smaller batch-size for Deberta-base.

To see if only headline is enough or we need article text as well to perform this task, we repeat the previous experiment with a slight change. Instead of only PostText now, we passed PostText and targetParagraphs separated by [SEP] token to the deberta and electra tokenizers to perform fine-tuning on Clickbait data. The training was done with the same hyper parameters as stated above except batch size being even smaller now as we are also using article now.

3.2.2 Headline Generation

We propose a novel method that does headline generation/article summarization using of-the-shelf T5-base on the entire dataset with a sequence length of 512. Then SentenceTransformer is used to generate 384-dimensional Embeddings of GT and Generated headlines. To find the similarity between GT and generated headlines, we use Cosine Similarity. We have assumed that a generated title is less likely to be Clickbait and hence if the original title was Clickbait then the Similarity between these two embeddings should be less. On the other hand, if the original title was non-clickbait so as the generated one, we will have a higher cosine similarity value. Thus, we might find a threshold for classifying valid and test data based on similarity values. This threshold for classification has been decided based on the Train sets' 'non-clickbait' chunk. Mean, the median of similarity values on the train set and trial-and-error have been performed to get the best threshold.



Lowest Cosine Similarity (PostText is GT headline)

	postText	generatedPostText	cosineSimilarity
51	huh.	Ivanka Trump on Syrian Refugees in US	-0.005111
28	ick	A Woman Arrested for Shoving Used Maxipad in A...	-0.004585
55	It's not over.	Oregon couple fined $135,000 for refusing to b...	-0.004046

Highest Cosine Similarity

	postText	generatedPostText	cosineSimilarity
9	How Well Do You Remember Chapter One Of "Stran...	How Well Do You Remember Chapter One Of "Stran...	0.999999
0	To save the Avon lady, the crockpot had to go	To Save the Avon Lady, the Crockpot had to go	1.000000
6	Can You Pass This Lie Detector Test?	Can You Pass This Lie Detector Test?	1.000000

Good enough generated headlines

	postText	generatedPostText	cosineSimilarity
205	How the Hubble Space Telescope changed the Uni...	Hubble Space Telescope	0.800162
4	This is what keeps #Mumbai's seashore so pollu...	Mumbai - The Most Polluted Sea Shore in the World	0.800398
46	Watch the full pre-credits 'Deadpool 2' teaser...	Deadpool 2 Teaser	0.800813

Figure 2: Generated Headline using of-the-shelf T5-base model

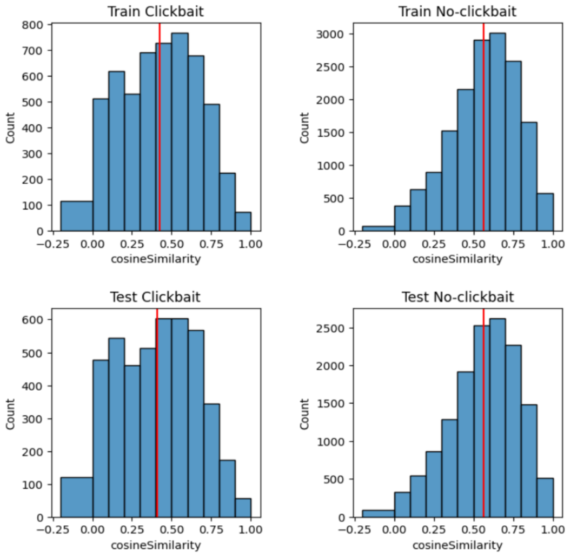


Figure 3: Histogram of Cosine sim on Clickbait and Non-clickbait Train and Test set

Original PostText	Article (starting line)	Generated Headline (of-the-shelf)	Generated Headline (fine-tune)	Class
Despite the ‘Yuck Factor,’ Leeches Are Big in Russian Medicine	MOSCOW — They are small as physician assistants go, about two inches long, and slithery. They wiggle	Leeches — Yes, Leeches!	Leeches — yes, leeches — are still widely prescribed in Russian medicine	no-clickbait
The top 10 best selling comic books of all time	More By Brian Prowse-Gany In celebration of National Superhero Day, Yahoo News counts down	Top 10 Comic Books of All Time	The Top 10 Marvel Comic Books of All Time	clickbait
this is good	President Donald Trump has appointed the pro-life advocate and former president of Americans	Trump Appoints Pro-Life Advocate Dr. Charmaine Yoest to Assistant Secretary of Public Affairs	Trump appoints Pro-Life advocate Charmaine Yoest to be Assistant Secretary of Public Affairs	clickbait
Bitcoin is spiking	Jonathan Garber, Business Insider 11.04.2017, 19:24 0 facebook linkedin twitter email print Bitcoin	Bitcoin is up 1.3% at $1,227 a coin as traders move into the cryptocurrency	Bitcoin is up 1.3% at $1,227<unk	no-clickbait

Table 1: Qualitative samples of Generated Headlines

The figure 2 shows some of the examples of original and generated headline and their cosine similarity scores. When we get low cosine similarity it is mainly when the original headline is clickbait. We also received nearly the same generated headline as the original sometimes and the score was nearly 1 for them.

Our assumption was to have high similarity between GT and Generated headlines for the non-clickbait articles and lower similarity for the clickbait articles. The achieved mean similarity for non-clickbait articles in the train set is 0.57 and 0.42 for non-clickbait. The same has been observed in the test set. From figure 3 we can infer that the plots are left-skewed for non-clickbait samples which supports our assumption but, finding a good classification threshold is not possible as the clickbait samples distribution is not right-skewed.

Since the performance of generated headlines from the off-the-shelf model was not up to the mark we decided to finetune a T5 model on our task of clickbait detection. We used the non-clickbait articles as the given text and the headline as the ground truth to finetune the model. We evaluated the Rouge-L score and used the best model to repeat the same steps explained above to see if we gain any further improvements.

Table 1 shows some qualitative results comparing the original headline, article, generated headline from the off-the-shelf model, and generated headline from the fine-tuned model with their original class. We can infer that fine-tuned model does generate better (less clickbaity) headlines which results in better accuracy as compared to the off-the-shelf model.

4 Dataset and Evaluation

4.1 Dataset

We used the Webis Clickbait Corpus 2017 (Webis-Clickbait-17) by Potthast et al. (2018) which comprises a total of 40,976 Twitter posts from 27 major US news publishers. In addition to the posts, information about the articles linked in the posts is included. The posts had been published between November 2016 and June 2017. To avoid publisher and topical biases, a maximum of ten posts per day and publisher were sampled. All posts were annotated on a 4-point scale [not click baiting (0.0), slightly click baiting (0.33), considerably click baiting (0.66), heavily click baiting (1.0)] by five annotators from Amazon Mechanical Turk. Using these labels it is decided whether the post is clickbait or not. A total of 9,276 posts are considered clickbait by the majority of annotators. The corpus is divided into three logical parts, training, validation and a test dataset.

Dataset	# Samples
Train	19538
Valid	2459
Test	18979

Table 2: Sample Count in Datasets

	Clickbait		No-ClickBait	
	#Count	%	#Count	%
Train	4761	24.37	14777	75.63
Valid	762	3.9	1697	96.1
Test	4515	23.11	14464	76.89

Table 3: Percentage Split of Each Dataset

As we can see the sample count of the train, validation, and test set is given in table 2. The table 3 shows the internal percentages of how much percent of each dataset is labeled as clickbait and non-clickbait. As we can analyze the validation dataset had about 3.9% of data as clickbait which is quite low as compared to others resulting in different metric values in the tests.

To visualize the dataset distribution, we plotted 2D TSNE of Train, Valid, and Test sets 'PostText Embedding' received from SentenceTransformer with the original dimension being 384. From figure 4 we can infer that all the sets belong to the same distribution and is highly dense.

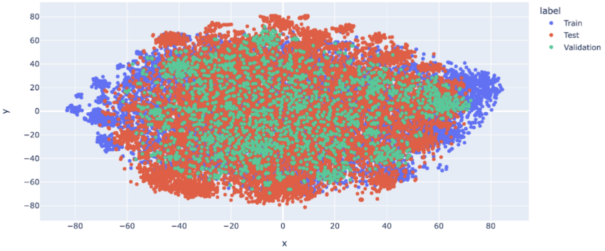


Figure 4: Visulaizing Train, Valid, Test set

Furthermore, to visualize clickbait vs non-clickbait samples' embeddings, we plotted similar 2D TSNE figure 5 on the Train set, we can see that there is not a clear distinction between the two classes making this a difficult classification problem.

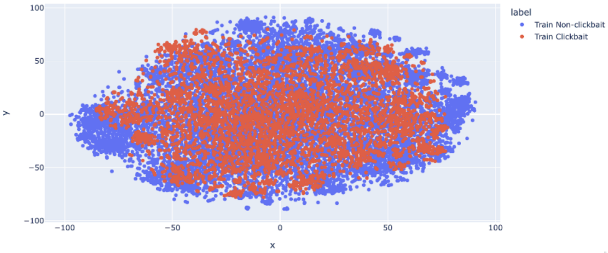
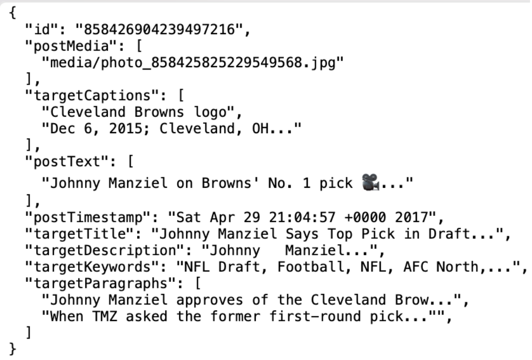


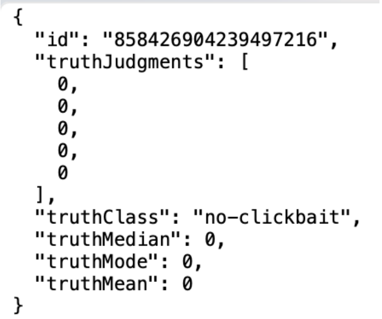
Figure 5: Clickbait and Non-clickbait Train Samples

Figure 6 is a single input sample (with truncated text for visualization purposes only). Its corresponding label in a JSON format is in Figure 7. The input contains post text, target captions, target titles, target descriptions, target paragraphs, etc. If all of these, a combination of these or only one of these can detect a headline as Clickbait or not is what we would try finding. The label file has truth judgments on 5 aspects as well as truth class and its mean, median, mode. Thus, can be used as either a Classification Task or as a Regression Task.



```
{
  "id": "858426904239497216",
  "postMedia": [
    "media/photo_858425825229549568.jpg"
  ],
  "targetCaptions": [
    "Cleveland Browns logo",
    "Dec 6, 2015; Cleveland, OH..."
  ],
  "postText": [
    "Johnny Manziel on Browns' No. 1 pick 🎥..."
  ],
  "postTimestamp": "Sat Apr 29 21:04:57 +0000 2017",
  "targetTitle": "Johnny Manziel Says Top Pick in Draft...",
  "targetDescription": "Johnny   Manziel...",
  "targetKeywords": "NFL Draft, Football, NFL, AFC North,...",
  "targetParagraphs": [
    "Johnny Manziel approves of the Cleveland Brow...",
    "When TMZ asked the former first-round pick..."",
  ]
}
```

Figure 6: Input sample (with truncated text)



```
{
  "id": "858426904239497216",
  "truthJudgments": [
    0,
    0,
    0,
    0,
    0
  ],
  "truthClass": "no-clickbait",
  "truthMedian": 0,
  "truthMode": 0,
  "truthMean": 0
}
```

Figure 7: Corresponding Label

4.2 Evaluation Metric

For the Evaluation metrics, we use the Accuracy Score and weighted F1 score because of the class imbalance for judging the models on how well they classify the data. F1 score is defined as the harmonic mean of the precision and recall and is a measure of accuracy.

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

For the Headline generation approach, we fine-tuned the model and select the best model based on the Rouge Score, Lin (2004).ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It uses the summarized text which is generated by the model and the ground truth text to evaluate how well the summarization task is being done. We particularly use the Rouge-L variant which is the Longest Common Subsequence based statistic.

5 Experiments

We performed fine-tuning of transformer-based models Electra-base and Deberta-base from Huggingface using Pytorch. Table 4 contains all the different values we tried for each of the models. The best results we got were with 3 Epochs, 2e-5

Learning Rate, 0.3 Dropout, 0.05 Weight Decay, and 128 Max Sequence Length. The selection of Batch Size was dependent on the memory limit from 8 to 64.

Hyperparamete	Values Tried
# Epochs	2, 3, 5, 10, 15
Learning Rate	1e-5, 1e-3, 5e-5, 2e-5
Dropout	0.1, 0.3, 0.5, None
Weight decay	0.01, 0.03, 0.05, 0.1, None
Seq Length	128, 256, 512

Table 4: Hyperparamete Tuning

Once we fixed the hyperparameters the next step was to find the best classification threshold. The logits we got from Electra or Deberta can be classified into 'clickbait' or 'no-clickbait' classes by taking argmax of the sigmoid output which in other words is taking threshold as 0.5. As we had imbalanced classes we thought of experimenting with different threshold values as shown in table 5. Which we simply removed one of the columns and applied a threshold on another column to classify. But, the best result we got on the validation set was with a threshold of 0.5 only. The same threshold we used on the test set later to get test accuracy and weighted F1 score.

		Validation	
Model	Classification Threshold	Acc	Weighted F1 Score
Electra	0.1	0.743	0.805
	0.2	0.763	0.815
	0.25	0.763	0.815
	0.3	0.764	0.809
	0.5	**0.778**	**0.855**
	0.6	0.769	0.814
	0.8	0.742	0.791
Deberta	0.25	0.762	0.814
	0.5	**0.782**	**0.859**

Table 5: Selecting the best Classification Threshold based on Weighted F1 Score on Val set (Headline-Variant)

To further evaluate our supervised models, we plotted the Precision-Recall curve as shown in figure 8. It is desired that the algorithm should have both high precision, and high recall. However, like most machine learning algorithms we can see a trade-off between the two here. We plotted such curves for different hyper params and thresholds and it help select the best model with the highest area under the PR curve.

These were all the experiments we did on the fine-tuning-based models. For the headline generation task, we notice the Rouge-L score at each epoch while fine-tuning T5-base for 10 epochs. We received the best score at epoch 3 which was also higher than the of-the-shelf model's rouge-L F1 score. This proved that fine-tuned model is better than using the off-the-shelf model as shown in table 6.

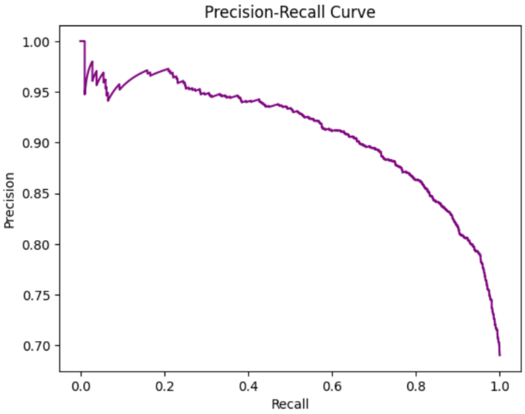


Figure 8: Precision Recall Curve on Electra-base

Model	Epoch	Rouge-L F1
Fine-tune T5-base	0	0.256
	1	0.251
	2	**0.259**
	3	0.257
Of-the-shelf T5-base	-	0.244

Table 6: Selecting the best fine-tuned T5-base model

Finally, to get the best cosine similarity thresholds, we plotted histograms as shown in figure 3 on the train, valid, test sets. We tried taking the Mean, the median of similarity values on the train set, and trial-and-error with other values to get the best threshold. For off-the-shelf T5 we got a good result with a threshold of 0.32 and on the fine-tuned model at 0.17.

The code for this project can be accessed at https://github.com/charmichokshi/ClickbaitDetector

6 Results and Discussion

6.1 Result

The Results of our experiments are shown in Table 7. The table's Feature column describe the data we

				Validation		Test	
Model	**Feature (H: Headline A: article)**	**Batch Size**	**Similarity Threshold**	**Accuracy**	**Weighted F1 Score**	**Accuracy**	**Weighted F1 Score**
SVM	H	-	-	72.4	0.83	81.6	0.889
XGBoost	H	-	-	71.1	0.824	78.7	0.875
Electra-base	H	64	-	77.79	0.855	86.73	0.914
Deberta-base	H	32	-	78.24	0.859	**87.51**	**0.92**
Electra-base	H+A	32	-	79.42	0.857	83.32	0.885
Deberta-base	H+A	16	-	79.3	0.864	87.46	0.918
T5-base Of-the-shelf	A	-	0.32	66.86	0.782	74.68	0.839
T5-base Fine-tuned	A	8	0.17	68.64	0.808	77.93	0.869

Table 7: Accuracy and Weighted F1 Score on Validation and Test set

used for that experiment. H is just headline and A is article. We reported two metrics Weighted F1 score and Accuracy for comparing different cases. In the last two cases where we used headline generation we also implemented a thresholding for the similarity values to get a better classifier.

In our results, the classical ML approaches reached about 81.6% accuracy level in the case of SVMs while the XGBoost reached just 78.7% on the test set. In the Deep Learning based approaches, the best result is achieved in the case of headline-only fine-tuning for Deberta with the accuracy levels reaching 87.51% on the test set closely followed by Headline and Article both combined as input which reaches 87.46% accuracy. Our headline generation approach performed poorly and was able to reach only 77.93% accuracy on the test set after fine-tuning the train set and thresholding the similarity at 0.17. The results of both Headline and Headline with article fine-tuning perform better than the current best models in Indurthi et al. (2020) which achieve about 85.7% on the test set.

6.2 Discussion

While analyzing the results we see that in most cases the performance of models is as expected with fine-tuning the transformer models on the data. But for the case of Headline Generation, the models seem to work poorly even after fine-tuning the task. Our analysis from figure 2 that in the case of non-clickbait the graphs are skewed which gives the similarity a kind of polarity which is good and can help in differentiating from the clickbait text. But, the same for clickbait is not true. The graph seems to be spread out evenly and covers quite a lot of range. There seems to be a very little amount of polarity when it comes to similarity. This results in an overlap between the ranges of non-clickbait and clickbait data leading to poor results for accuracy. Even after performing thresholding and fine-tuning we were able to push the accuracy a bit further but the underlying characteristic of data is creating issues for this hypothesis. Another issue that we found was the inconsistency of results for the validation set. The values for the test set and validation set differ by quite a significant margin. This can be attributed to the differences in the size of the validation set as compared to the test and train set. Also, another major issue is that the percentage of clickbaits in the validation set is quite low at 3.9% compared to 24% in the other two sets which can be a source of the difference.

7 Conclusion

Overall, we explored ML and DL approaches for language classification task and also explore the text summarization field. We fine-tuned models on only headlines to predict if the article is clickbait or not worked well as compared to the same models fine-tuned on headline and article. We were assuming that using article text might improve that performance but it did not. The off-the-shelf headline generation and our hypothesis of using cosine similarity worked but not very well as there was significant overlap between the non-clickbait and clickbait headline embedding and their similarity scores. We improved this approach by fine-tuning the title generation model and got a better threshold for separation but it is still less than Deberta fine-tuned on the headline alone.

Talking about the use case of clickbait detection, it can have a wider use case as an add-on extension on web browsers showing if the headline is clickbait and whether the user should click on it or not. As the results indicate providing the headline is sufficient to classify the clickbaits it can help consumers in real-time to avoid shady websites that lure people. For future work, we plan on doing Clickbait spoiling, a task in SemEval-2023 which is a step further. It incorporates finding the answers to intriguing and luring questions which are used as clickbaits by getting the answer from the article text for the user or simply spoiling it because the user doesn't have to click on the link anymore.

8 Code

The code for this project can be accessed at https://github.com/charmichokshi/ClickbaitDetector. Please refer to the ReadMe file for more details.

9 Contributions

Both the members worked equally on data preprocessing, cleaning, report writing, and presenting throughout the term. Modeling of classical-ML approaches, Electra-base (H), and fine-tuning headline generation was done by Sandeep. Modeling of Electra-base (H+A), Deberta-base (H, H+A), and off-the-shelf headline generation was done by Charmi. Overall, we believe both members contributed equally to the project.

10 Acknowledgment

We like to thank the Google Developers Expert program for providing the GCP credits which helped us train larger language models with ease. We also thank our mentor for the project Xing Han Lu for the numerous fruitful discussions and for guiding us throughout the project.

References

Amol Agrawal. 2016. Clickbait detection using deep learning. In *2016 2nd international conference on next generation computing technologies (NGCT)*, pages 268–272. IEEE.

Ankesh Anand, Tanmoy Chakraborty, and Noseong Park. 2017. We used neural networks to detect clickbaits: You won't believe what happened next! In *European Conference on Information Retrieval*, pages 541–547. Springer.

Arun Babu, Annie Liu, and Jordan Zhang. 2017. New updates to reduce clickbait headlines. *Facebook Newsroom*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Abhijnan Chakraborty, Bhargavi Paranjape, Sourya Kakarla, and Niloy Ganguly. 2016. Stop clickbait: Detecting and preventing clickbaits in online news media. In *2016 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM)*, pages 9–16. IEEE.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Vijayasaradhi Indurthi, Bakhtiyar Syed, Manish Gupta, and Vasudeva Varma. 2020. Predicting clickbait strength in online social media. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4835–4846.

Sawinder Kaur, Parteek Kumar, and Ponnurangam Kumaraguru. 2020. Detecting clickbaits using two-phase hybrid cnn-lstm biterm model. *Expert Systems with Applications*, 151:113350.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

George Loewenstein. 1994. The psychology of curiosity: A review and reinterpretation. *Psychological bulletin*, 116(1):75.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Martin Potthast, Tim Gollub, Matthias Hagen, and Benno Stein. 2018. The clickbait challenge 2017: Towards a regression model for clickbait strength. *CoRR*, abs/1812.10847.

Martin Potthast, Sebastian Köpsel, Benno Stein, and Matthias Hagen. 2016a. Clickbait detection. In *European Conference on Information Retrieval*.

Martin Potthast, Sebastian Köpsel, Benno Stein, and Matthias Hagen. 2016b. Clickbait detection. In *European conference on information retrieval*, pages 810–817. Springer.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Praboda Rajapaksha, Reza Farahbakhsh, and Noel Crespi. 2021. Bert, xlnet or roberta: The best transfer learning model to detect clickbaits. *IEEE Access*, 9:154704–154716.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Yiwei Zhou. 2017. Clickbait detection in tweets using self-attentive network. *arXiv preprint arXiv:1710.05364*.