Is that headline Clickbait?

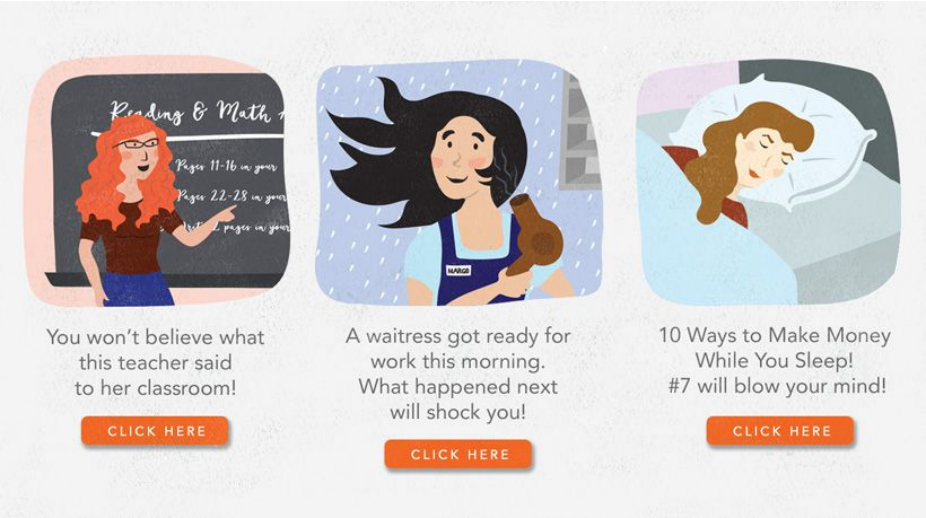
A Transformer-based News Clickbait Detector

By: Charmi Chokshi
Sandeep Kumar

Mentor: Xing Han Lu

What?

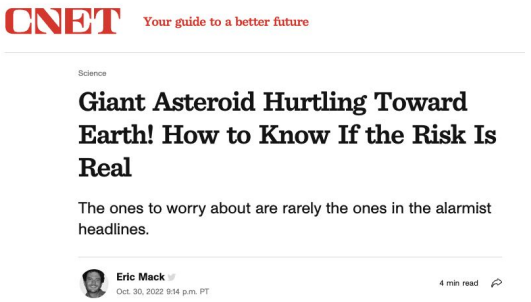
- A headline is Clickbaiting when it is,
- Arousing curiosity instead of providing informative summaries
- Mostly misleading to their reader
- Generated just to increase user engagements and 'clicks' in posts



Why?

- To promote responsible journalism
  - Recent example below of a celebrity calling out a publication for using his interview as clickbait
- To stop the spread of misleading information in today's information world
  - Recent example on the right showing an actual note VS how print and digital media take it over

Tyler aired the publication out for being thirsty for clicks. He said, “I guess people write these kinds of headlines for clickbait. It’s sad. I didn’t call out Janet and Jill, and they’re not holding up another Why Did I Get Married.” According to Tyler, he doesn’t even have a script for the third installment, just an idea for a storyline.



How?

1. Classical ML-based approaches (SVM, XGBoost)
2. Fine-tune Transformers on Headline (Electra, Deberta)
3. Fine-tune Transformers on Headline and Article (Electra, Deberta)
4. Generate Headline from given Article and find its Embeddings Similarity with the original Headline Embedding. Use this similarity to compute threshold for Classification Task (T5-base of-the-shelf, SentenceTransformer)
5. Fine-tune Transformers on the Generated Headlines from above (Electra, Deberta)
6. [WIP] Fine-tune Headline Generation Task (T5-base)

Dataset

Dataset

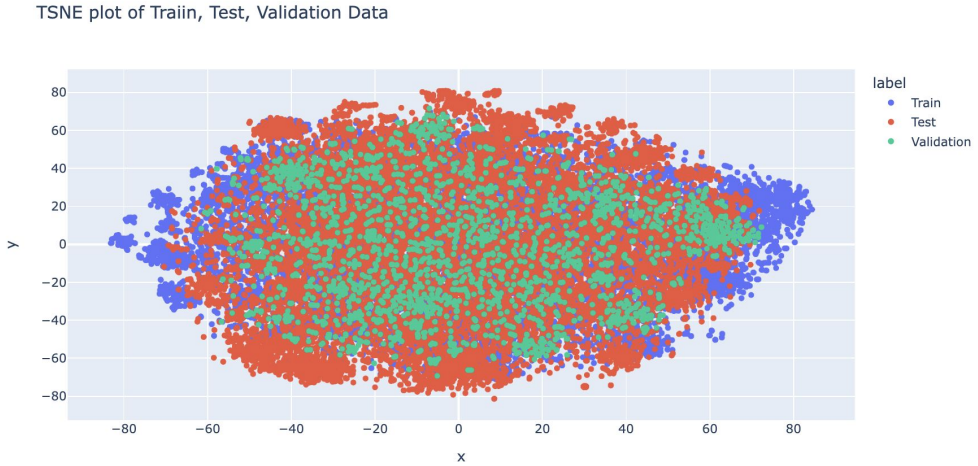
- **Webis-Clickbait-17**: the Webis Clickbait Corpus 2017 comprises a total of 40,976 Twitter posts from 27 major US news publishers

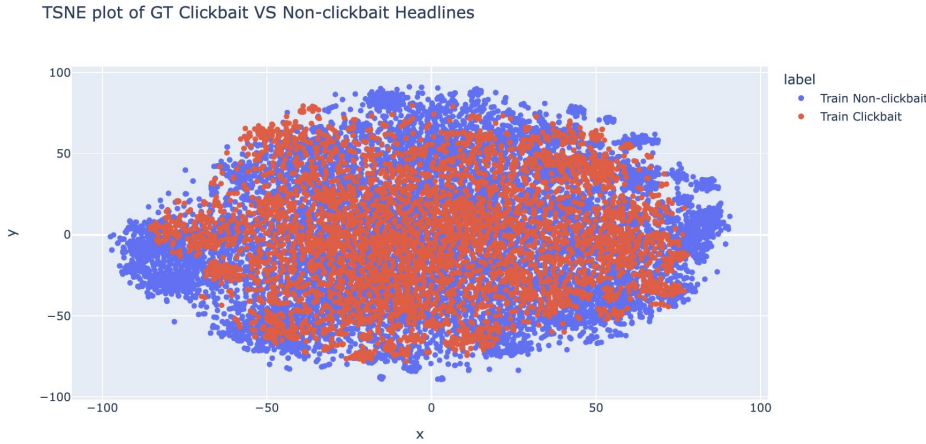
Dataset	# Samples
Train	19538
Valid	2459
Test	18979

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

Dataset Distribution

- Visualizing Train, Validation and Test Set embeddings (from SentenceTransformer)





Data and Label Sample

- Each sample contains article headline, passages, keywords, captions, etc.
- Annotated on a 4-point scale and their class


```
{
  "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..."",
  ]
}
```


Data Sample


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


Corresponding Label

Methods

Classical ML-based Approaches

- We have used SVM and XGBoost classifiers for clickbait classification task
- The SVM utilizes RBF (Radial Basis Function) kernel for computation
- The XGBoost had 150 estimators and maximum depth allowed was 3
- SVM achieves nearly 81% accuracy and XGBoost about 79% thus, SVM performs slightly better than the XGBoost

Fine-tuning on Headline feature

- We used Transformer-based Classification architectures: Electra-base and Deberta-base
- Processing on input text such as removing special characters, removing Null values and unwanted columns has been done
- We get Embedding of *PostText* using the respective tokenizers and fine-tune the models with the below setup

Modelling Setup	
# Epochs	3
Learning Rate	2.00E-05
Dropout	0.3
Weight decay	0.05
Seq Length	128

Fine-tuning on Headline and Article feature

- 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
- Insted of only *PostText* now, we pass *PostText* and *targetParagraphs* separated by [SEP] token to the deberta and electra tokenizers to perform fine-tuning on Clickbait data

Classification based on on-the-shelf Headline Generation

- We propose a novel method which does headline generation / article summarization using T5-base on the entire dataset with sequence length 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
- Threshold for classification has been decided based on the Train sets' 'non-clickbait' chunk and is been tested on the entire test set. Mean, median, and trial-and-error has been performed to get the best threshold

Examples of Generated Headline

- 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

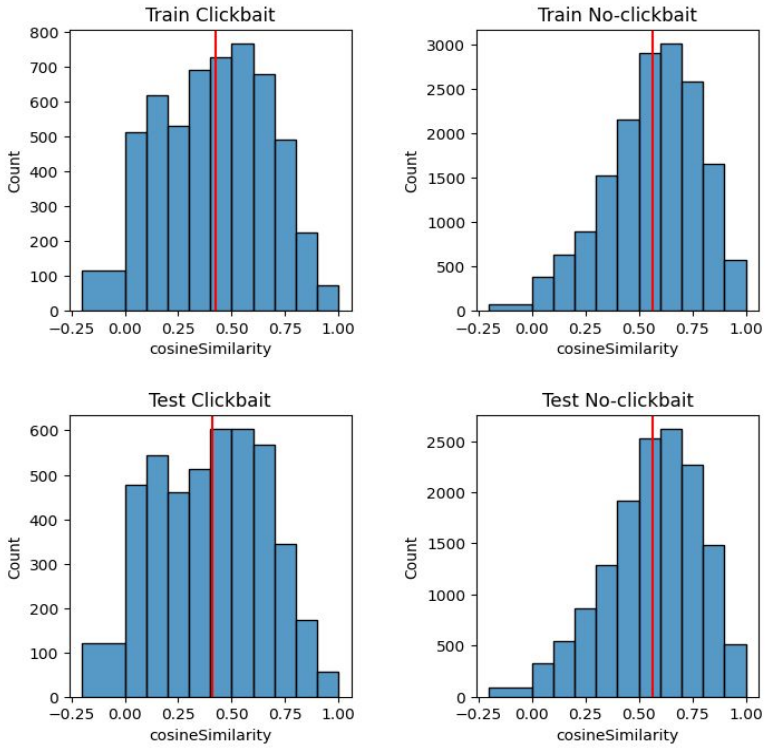
	postText	generatedPostText	cosineSimilarity
9	How Well Do You Remember Chapter One Of "Stran...	How Well Do You Remember Chapter One Of "Stran...	0.999999
0	To save the Avon lady, the crockpot had to go	To Save the Avon Lady, the Crockpot had to go	1.000000
6	Can You Pass This Lie Detector Test?	Can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

Deciding the Classification Threshold

- Our assumption was to have high similarity between GT and Generated headlines for the non-clickbait articles and lower similarity for the click-bait articles
- The achieved mean similarity for non-clickbait articles in train set is 0.57 and 0.42 for non-clickbait. The same has been observed in the test set
- But finding a good classification threshold is not possible as the clickbait samples distribution is not right-skewed



Fine-tuning Headline Generation Task

- Since the performance of generated headlines from 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 headline as the ground truth to finetune the model
- We evaluated the rouge-L score and will use the best model to repeat the work done for off the shelf model to see if we gain any further improvements
- Working on the Classification part

Result

Result

- All the models have low validation Accuracy and F1 score as compared to test
- Transformer models worked well when using only headline feature

			Validation		Test	
Model	**Feature**	**Batch Size / Threshold**	**Accuracy**	**F1**	**Accuracy**	**F1**
SVM	Headline	-	72.4	0.83	81.6	0.889
XGBoost	Headline	-	71.1	0.824	78.7	0.875
Electra-base	Headline	64	77.79	0.855	86.73	0.914
Deberta-base	Headline	32	78.24	0.859	87.51	0.92
Electra-base	Headline + Article	32	79.42	0.857	83.32	0.885
Deberta-base	Headline + Article	16	79.3	0.864	87.46	0.918
T5-base of-the-shelf	Article	Cosine Sim Thresh: 0.32	66.86	0.782	74.68	0.839

Conclusion

- Overall in the results we have till now the fine-tuned models on only headlines to predict if the article is clickbait or not worked well
- 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 are further improving this approach by fine-tuning the title generation model and probably we would be able to get a better threshold for seperation

Thank You!

Questions?