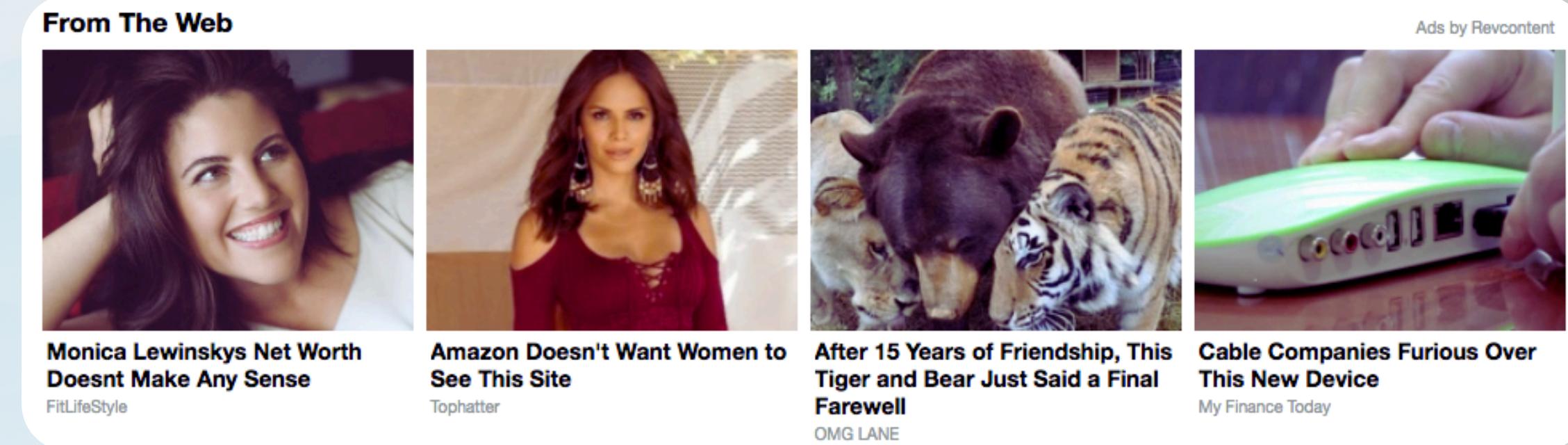


Clickbait Spoiler Detection and Generation

Poluri Rishi Manoj (CS21BTECH11045)

Kota Vignan (CS21BTECH11029)

- Group 17



Problem Statement

- Clickbaits are texts containing sensationalized terms that aim at raising the curiosity of the readers, and lure them to click on a hyperlink. These are common in news headlines, social media posts etc.
- These usually fall short of readers' expectations, wasting their time and causing disappointment.

Due to the deceptive and misleading nature of the content often presented, clickbait is generally considered harmful. Thus addressing the clickbait issue is a crucial concern.

Solution:

- The generation of spoilers, i.e. short texts that neutralize clickbait by providing information that satisfies the curiosity induced by it.

Work Done so far ...

- Implementing the approach given in the paper
 - This involves two steps: 1) finetuning a model for this particular task of spoiler detection, 2) post-hoc ensembling for outputs generated on same model for different seeds
 - Finetuning - used T5-base with huggingface
 - Post-hoc - run T5-base for different seeds (43, 45, 46, 47, 48) and take the best output on each candidate using edit-distance.
- Finalising an approach for project
 - Referred different research papers
 - Discussed ideas and fixed on an approach
- Few initial trials on different models for specific tasks like spoiler type classification
 - Approach involves spoiler type classification
 - Worked out on few different models to find that can best fit to the task

Work Division

Rishi Manoj

CS21BTECH11045

- Trying out few models for finetuning
- Implementing and evaluating on the model that gave good results
- Referring to research papers and resources for a different approach
- Involving in discussion

Vignan

CS21BTECH11029

- Implementing post-hoc ensembling
- Evaluating the model and getting results
- Referring to resources for ideas
- Involving in discussion

Code Presentation

Results

Single

```
You should probably train this model on a downstream task to be able to use it for predictions and I  
BERTScore: {'precision': 0.8847038745880127, 'recall': 0.8751260042190552, 'f1': 0.8792693018913269}  
BLEU-4 Score: 0.25643600757035584  
TestLoss: 0.22385539564490317
```

Ensemble

```
TestLoss: 0.2194286795258522  
BERTScore: {'precision': 0.8899759650230408, 'recall': 0.8760814070701599, 'f1': 0.8823524713516235}  
BLEU-4 Score: 0.3095721068976759
```

Prediction: Farrah Abraham
Reference: Farrah Abraham

Prediction: Kim Kardashian Checking Her Makeup
Reference: Kim Kardashian Checking Her Makeup

```
Prediction: 1. Your relationship difficulties 2. Your detailed financial state -- or financial problems 3. Your legal woes  
Reference: 1. Your relationship difficulties 2. Your detailed financial state -- or financial problems 3. Your legal woes
```

Comments on Results

- Implementing single model (T5-base) gave:
 - BERTScore - ~0.879
 - Bleu - ~0.256
- Implementing ensemble gave:
 - BERTScore - ~0.882
 - Bleu - ~0.309
- Using ensemble gave better results than using a single model, which is better for spoiler generation.
- Observing predictions of the model, the predictions doesn't generate better spoilers for passage and multipart.

Our Approach

- Spoiler Type Classification
 - Classify the appropriate spoiler type (phrase, passage, or multi-part)
 - Approach: Develop a classifier model that determines which spoiler type best suits the input
- Spoiler Generation
 - After obtaining the spoiler type, include it as part of the prompt or input sequence for each sample and finetune a model to generate spoilers
- Ensembling
 - Generate outputs for different models and taking ensemble from them

Currently Working on

- Spoiler type classification
 - Model Selection: Exploring different classification models leveraging their capabilities in understanding contextual nuances.
 - Training and evaluate their performance using metrics such as accuracy, precision, recall, and F1-score
 - Selecting which performs best for the data
- Work Division: Both of us currently working on this part which is major part of our approach.

Plan for upcoming weeks

- Completing spoiler type classification model implementation
- Finetuning model to generate spoiler given context and spoiler type as well, as input
 - Both combinely takes **(1 week)**
- Implementing - Ensembling few different models and evaluating **(1 week)**
- Report and final deliverables **(<1 week)**

If Possible ...

- **Condense information** from long articles for model input.
- Improves model performance by focusing on relevant information.
- This may involve filtering paragraphs and gaining advantage by feeding more focused information to the model.
- We'll be implementing this if time permits.

Thank You