

Final Project Proposal

Clickbait Vs. Informative Classifier for News Headlines

1. Problem Statement

This project will focus on classifying short news titles as either *clickbait* or *informative*. Language in clickbait headlines is designed to turn curiosity into clicks with sensational or exaggerated phrasing while informative headlines provide a more balanced summary of an article's content. The internet is littered with clickbait and the ability to effectively categorize an article accordingly could save users time and promote better information dissemination as a whole.

2. Approach

The central task is a supervised binary text classification. Given a headline, the algorithm should predict whether it is clickbait or informative. This will be done in two ways. First, using classical machine-learning techniques. Headlines will be turned vectorized and classified to build a baseline which can then yield interpretable results. This includes associating tokens to labels, either *clickbait* or *informative*. Second, a transformer-based model will perform the same task. This will be pre-trained, like BERT. Both models will be trained on the same training and validation splits to allow for fair comparison.

3. Evaluation

Performance will be evaluated on a test-set that is not used during training or tuning. Primary metrics include accuracy, precision, and F1 score. A confusion matrix will be used to visualize error patterns. For the classical model, the highest-weighted features will be inspected for insight into what linguistic patterns or words generate one output or the other. The transformer model can be evaluated against specific headline inputs to show which headlines it handles better than others.

4. Tools, Techniques and Datasets

- a. Data Handling: pandas, numpy, scikit-learn, Hugging Face *transformers* library
- b. Visualization: matplotlib for confusion matrices
- c. Datasets: Kaggle clickbait dataset
(<https://www.kaggle.com/datasets/amananandrai/clickbait-dataset>)

5. Core Goals

- a. Performant clickbait classifier: implement models that demonstrate proper classification.

- b. Model Comparison: Show performance for classical vs. transformer models to offer both quantitative and qualitative explanations.
- c. Model error analysis: investigate and describe where models fall short and provide some insight into why this might happen based on model characteristics and data.

6. Stretch Goals

- a. Minimal web interface (e.g. Flask) where a user can paste a headline and see label (clickbait or informative) and model confidence score, visualizations associated with models.
- b. Display most influential tokens for the prediction for the classical model.
- c. Batch processing: CSV upload with headlines to output labels for each