Rex Sheikh
CSPB 3832: Natural Language Processing
11-18-2025

Final Project Interim Report

Project Summary

This project focuses on classifying short news headlines as either *clickbait* or *informative*. Clickbait headlines are written to maximize curiosity and clicks through sensational or vague language, while informative headlines aim to provide a concise and accurate summary of the article content. Since the project proposal, the core natural language processing task is still binary text classification. The goal of this report is first to describe data sources, both new and previously reported. The approach with respect to this data will follow along with current progress and next steps to reach project completion. The final section poses some open questions that were generated since beginning this project.

Data

a. Kaggle Clickbait Dataset
   i. Source
   ii. Size/Structure: 1.8MB, 32000 entries with format Headline, 0 (not clickbait) or 1 (clickbait)
   iii. Discussion: This dataset is split in half where the first half is classified as clickbait and the second is not. An example of clickbait is “Are you more Walter White or Heisenberg?”. An informative headline is “Canadian-born actor Leslie Nielsen dies aged 84.” Quick inspection of the data matches my intuition for the binary classification. The dataset was initially hosted on Kaggle in November, 2024 and most of the articles do seem contemporary. It is somewhat concerning that there are no sources for these articles, though. At the very least it would be interesting to see which sources were more apt to publish one classification or another. See the “Progress” section for a discussion on whether or not a sample of these articles were found to be real or not.

b. Webis Clickbait Corpus 2017 (Webis-Clickbait-17)
   i. Source
   ii. Size/Structure: The corpus can be found at the above link under the title “clickbait17-train-170630.zip.” There are two main files, instances.jsonl and truth.jsonl. It’s size is 12.5MB, 19538 entries in JSON format for both instances and truth. Here is an example of an instance/truth pair.


```
{
  "truthJudgments": [
    1,
    1,
    1,
    1,
    1
  ],
  "truthMean": 1,
  "id": "858464162594172928",
  "truthClass": "clickbait",
  "truthMedian": 1,
  "truthMode": 1
}
```

```
{"postMedia": [], "postText": ["UK\u2019s response to modern slavery leaving victims
destitute while abusers go free"], "id": "858462320779026433", "targetCaptions":
["modern-slavery-rex.jpg"], "targetParagraphs": ["Thousands of modern slavery victims
have\u00a0not come forward, while others" "targetKeywords": "modern slavery, Department
For Work And Pensions, People Trafficking, Frank Field, Home News, UK, News",
"targetDescription": "\u201cInexcusable\u201d failures in the UK\u2019s system for
dealing with modern slavery are\u00a0leaving victims reduced to destitution while their
abusers go free because they are not adequately supported to testify against them, an
alarming report has warned."}
```


   iii. Discussion: The above link starts with a succinct description of the data: “The Webis Clickbait Corpus 2017 (Webis-Clickbait-17) comprises a total of 38,517 Twitter posts from 27 major US news publishers. In addition to the posts, information about the articles linked in the posts are included.

The posts had been published between November 2016 and June 2017." This dataset is a great resource for not only core project aims but for follow-on or related work too. For core goals, it's clear that it can be applied to the binary classification task by matching postText to clickbait scores.

c. News Clickbait Dataset
   i. Source
   ii. Size: 2 datasets. The first is from "Stop Clickbait: Detecting and Preventing Click baits in Online News Media" with 32,0000 entries. The second is from "Clickbait news detection dataset." Both have format title, news or clickbait in csv format.
   iii. Discussion: This dataset is very similar to the first with some improvements. The first dataset originates from "Stop Clickbait: Detecting and Preventing Click baits in Online News Media" while the second was featured in a popular competition on Kaggle. At the very least, they give the opportunity to compare performance differences between data.

Approach

In general, I will be starting with tools and techniques we learned earlier in the semester and progressing towards later techniques. I plan to keep the metrics the same across datasets and experiments. Key metrics will be captured via matplotlib.

1. Baseline Classical Model for all datasets, starting with Kaggle Clickbait Dataset (first data listed above)
   a. Goal: Establish a strong, interpretable baseline on the Kaggle Dataset
   b. Methods/Models
      i. Vectorization: bag-of-words, TF-IDF using unigrams/bigrams/n-grams
      ii. Algorithms: logistic regression and linear SVM.
      iii. Tools, Libraries: pandas, numpy, scikit-learn
      iv. Metrics: accuracy, precision, recall, Confusion matrix, ROC-AUC and PR-AUC, Per-class precision/recall and macro/micro averages
2. Model Refinement and Ablation Studies
   a. Goal: Improve/tune models for each dataset before moving to transformers
   b. Methods/Models
      i. Parameter tuning for logistic regression and linear SVM.
      ii. Ablation studies on feature sets
      iii. Data cleaning (stopwords at least).
      iv. Class weighting.
      v. Metrics: Same as above
3. Model Comparison
   a. Goal: Capture key metrics and data characteristics to give insight into performance and interesting results.
4. Transformer-based Model
   a. Goal: Fine-tune a pre-trained transformer on the same splits to compare directly against classical baselines
   b. Methods/Models
      i. `bert-base-uncased` (or a comparable transformer) via Hugging Face
      ii. `matplotlib` for evaluation plots
5. Comparison and Error Analysis
   a. Inspect high-confidence false positives and false negatives
   b. Look for any systematic patterns like word length, topic, keywords

Progress

1. My major point of progress so far has been consolidating code snippets to be used for this project. I've reviewed and began implementing the following:
    a. Logistic Regression using LogisticRegression.py (lecture code) and homework4.py.
    b. Metrics to include accuracy, precision, confusion matrix from hw6/problem3.py
    c. Transformers in Hugging face from hw9.

Next Steps

1. Integrate code snippets into the three data sets for both baseline classical models
2. Capture results and present them with matplotlib
3. Integrate code snippets into the three data sets for the transformer model
4. Outline performance differences between models
5. (Stretch) Web app using flask for user input, visualization suite

Questions

In putting this interim report together, I was curious how much effort it takes to assemble quality data and label accordingly. Especially for the second dataset (Webis Clickbait Corpus 2017 (Webis-Clickbait-17)) there were many, many contributors and separate responses for each truth response. Further, the last dataset's associated paper has half a dozen authors and and seems to have been a seriously time intensive effort. I had assumed that web crawling would be sufficient for some open source data but was surprised to see so many contributors in each case.