Rex Sheikh Final Project: CSPB 3832 Natural Language Processing 12.1.2025

Detecting Clickbait

Using natural language processing techniques to label headlines as either ‘clickbait’ or ‘news’ with classical and transformer-based methods.

Introduction and Background

Clickbait (noun) : something (such as a headline) designed to make readers want to click on a hyperlink especially when the link leads to content of dubious value or interest.

News (noun) : a report of recent events.

Misrepresentation, misinformation, and qualitatively bad information is rife on the modern internet. Many pieces hide behind article headlines with superficially alluring language designed purely for maximizing engagement. Often, their content falls short of what would be considered news. The core aim of this project is to leverage natural language techniques to uncover headline features associated with either clickbait or news. Classification could then be applied to mitigate adverse effects broadly. Comparison of different techniques across different datasets can also give insight into their strengths, weaknesses, and inner workings in general.

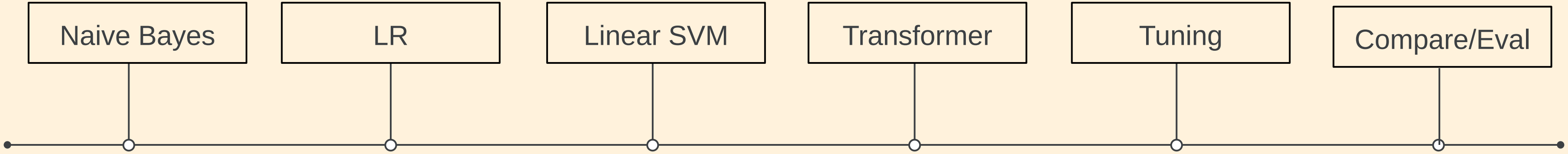
Problem Statement

Given curated datasets for training and testing, use both classical and transformer-based machine learning tools to quantitatively classify headlines as either clickbait or news.

Implied Tasks

- Load datasets, apply test and training split
- Choose models, determine baseline for each
- Choose performance metrics
- Craft experiments to refine model performance and extract key language features
- Visualize and compare results

Methodology



Naive Bayes

- Tokens are conditionally independent.
- Choose class whose word-likelihood best explains the token counts.
- nltk.NaiveBayes

LR

- Learn scoring features from TF-IDF
- Uses sigmoid to give more calibrated probability
- sklearn LogisticRegresion

Linear SVM

- Learn scoring features from TF-IDF
- Use linear hyperplane to separate labels with configurable margin.
- SGDClassifier

Transformer

- Uses self-attention to combine information across tokens.
- Captures word-order, phrase patterns, etc.
- ‘Bert-based-uncased’
- ‘distilbert-base-uncased’

Tuning

Compare/Eval

Baseline

- 80/20 train/test split
- Minimal preprocessing
- Vocabulary default set to 2000

Experiments

- Feature Addition/Ablation
- Class reweighting
- Pruning thresholds

Metrics

- Accuracy
- Precision (per-class, micro/macro)
- Recall (per-class, micro/macro)
- Confusion Matrix
- ROC-AUC/PR-AUC

Data/Materials

Kaggle Clickbait Dataset

- ➔ Source
- ➔ 32,000 entries with format headline/label in CSV format

Webis Clickbait Corpus

- ➔ Source
- ➔ 195389 entries in JSONL format split into instances.jsonl and truth.jsonl
- ➔ Match id entries to extract headline/truthClass pairs
- ➔ Entire text, additional media, and other fields available (related to future work/questions)

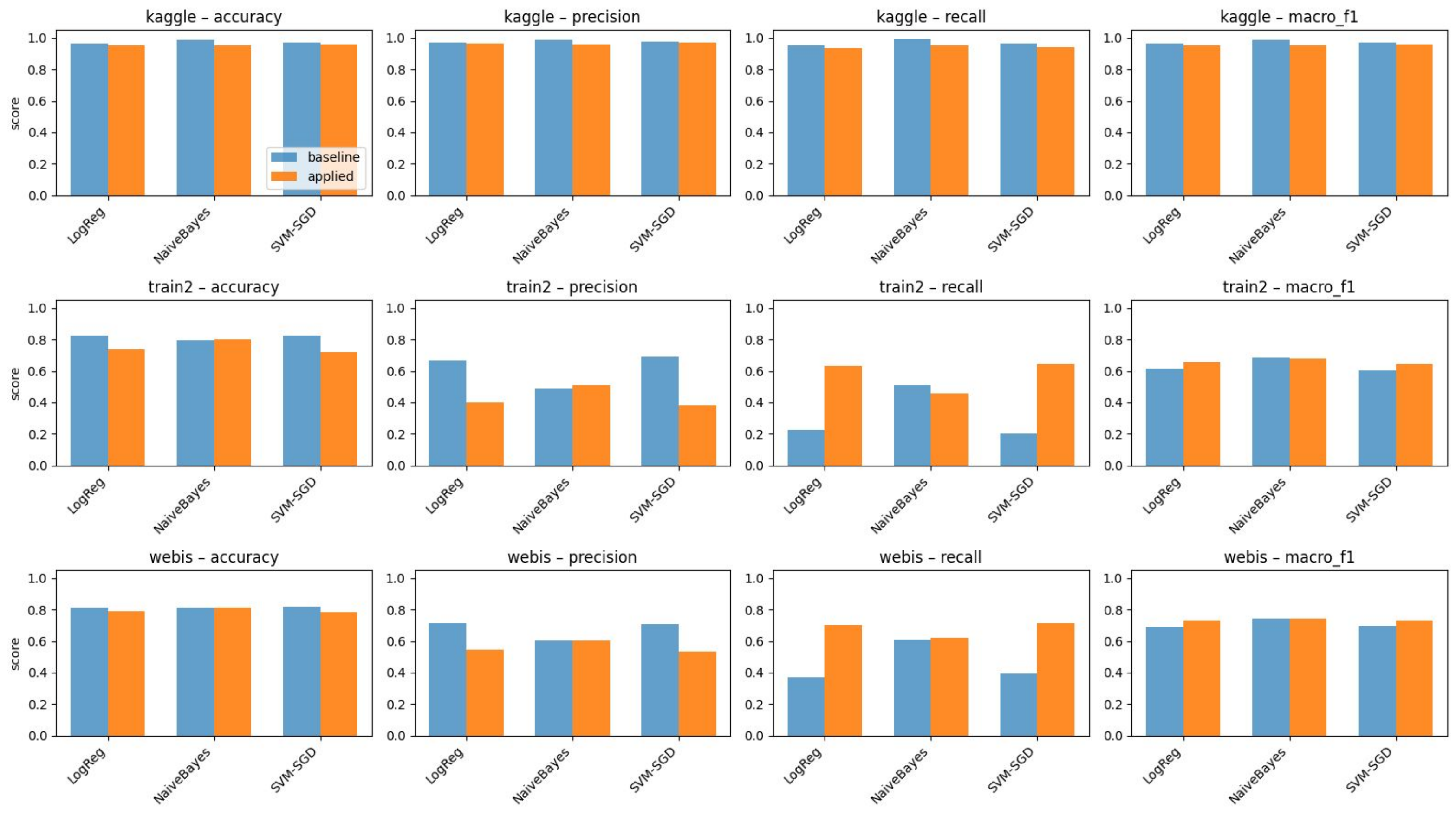
News Clickbait Dataset

- ➔ Source
- ➔ 2 CSV datasets, train1.csv appears identical to Kaggle Clickbait Dataset, only using train2.csv in experiments
- ➔ 32,000 entries in format headline/label

Hugging Face

- ➔ T4 GPU
- ➔ ‘distilbert-base-uncased’
- ➔ ‘bert-base-uncased’

Results - Metrics Classical (NB, LR, and SVM)



Results - Naive Bayes Top Tokens

Dataset	Config	Class Bias	Tokens (Top 3)
Kaggle	Baseline	non	in, for
		clickbait	Of
	Stopwords+Ignore	clickbait	Based, Things, Actually
Train2	Baseline	non	Breitbart, dies
		clickbait	Ent
	Stopwords+Ignore	non	Breitbart
		clickbait	Quiz, Products
Webis	Baseline	non	least, amid
		clickbait	ll
	Stopwords+Ignore	non	least, amid
		clickbait	ways

Results - LR Top Identifiers

Dataset	Config	Clickbait Cues (Top 3)	News Cues (Top 3)
Kaggle	Baseline	you, this, your	in, dies, says
	Stopwords+Ignore	people, actually, things	dies, wins, china
Train2	Baseline	you, these, this	breitbart, york, says
	Stopwords+Ignore	things, list, ent	breitbart, york, says
Webis	Baseline	this, why, these	says, trump, in
	Stopwords+Ignore	things, ways, quiz	says, trump, police

Results - SVM Top Identifiers

Dataset	Config	Clickbait Cues (Top 3)	News Cues (Top 3)
Kaggle	Baseline	you, this, your	says, knicks, dies
	Stopwords+Ignore	actually, know, things	dies, wins, china
Train2	Baseline	you, these, your	breitbart, york, cup
	Stopwords+Ignore	things, quiz, ways	breitbart, york, years
Webis	Baseline	this, these, why	says, trump, is
	Stopwords+Ignore	things, ways, quiz	trump, says, million

Results - Transformer

Dataset	Model (max_len / epochs)	Acc	Prec (pos=1)	Rec (pos=1)
Kaggle	BERT base (128 / 3)	0.99	0.989	0.99
	DistilBERT (64 / 2)	0.988	0.991	0.986
Train2	BERT base (128 / 3)	0.825	0.576	0.458
	DistilBERT (64 / 2)	0.831	0.623	0.392
Webis	BERT base (128 / 3)	0.857	0.715	0.652
	DistilBERT (64 / 2)	0.853	0.702	0.654

Conclusion

Dataset Insights

- ➔ Kaggle headlines: every model (Naive Bayes, TF-IDF linear, transformers) exceeds 0.95 accuracy, and the best hits ≈0.99 with balanced precision/recall, implying vocabulary/style differences alone flag clickbait.
- ➔ Train2 is heavily imbalanced (only ~20% clickbait) on the deterministic shuffle models consistently achieve high news precision but struggle to recall clickbait
- ➔ Webis sits between the two: ~0.81 accuracy with classical pipelines and up to ~0.86 with transformers, showing moderate difficulty and value in contextual modeling.

Model Insights

- ➔ Naive Bayes excels when tokens discriminate well. Stopwords can obscure other interesting examples.
- ➔ Logistic Regression and SVM-SGD capture stronger semantic cues than NB once stopwords are removed
- ➔ Transformers (BERT/DistilBERT) capture context beyond word frequency, yielding superior macro balance on difficult datasets (Train2/Webis). They also match classical scores on Kaggle; however, they remain sensitive to class imbalance (Train2 recall ~0.45–0.65)
- ➔ DistilBERT provides a good speed–accuracy trade-off: only minor degradation relative to BERT on Kaggle/Webis, though its reduced capacity leads to lower clickbait recall on Train2 when compared to full BERT.

Open Questions, Future Work

- *Does article content match with headline label?*
  - Article Body Inspection: Extend the labeling analysis.
  - Webis dataset would be a good start, already provides article text
- *Can article media (photos in particular) strengthen a labeling signal? Intuitively what photo features would correspond to either label?*
  - Article Media Inspection: Extend labeling analysis to see if photos corroborate clickbait or news.
- *What clickbait language features result in the most engagement?*
  - Once labeled, it would be interesting to inspect which clickbait articles drew the most engagement.
  - Could use multinomial classification where bins are assigned to the type of clickbait.
- *How can this binary classification be used in other news contexts?*
  - Bullish/Bearish (Financial News)
  - Left/Right (Political News)
- *What are some ways users can use and extend this project?*
  - Simple UI or Web Extension Tool: apply labels quickly to user copy-pasted headlines
- *How much do individual feature sets affect end results?*
  - The CLI allows for some fine-tuned experiments. Use this to investigate which features affect key metrics the most.