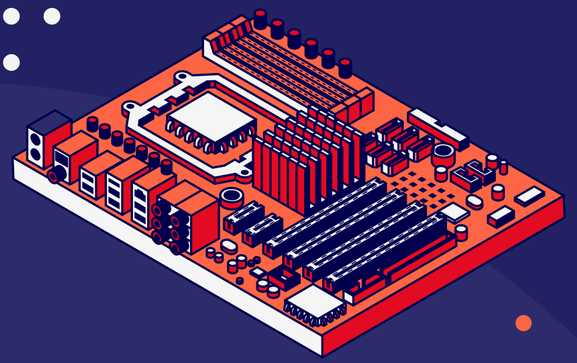
Using Context to Determine Sarcasm in Reddit Comments

Sandy Thomas



Background

- Sarcasm detection is an active challenge in NLP(Natural Language Processing)and computational Pragmatics
- Machines can have a hard time detecting sarcasm using traditional text-processing techniques because sarcastic language often contradicts the literal language meaning
- My project explores using:
- Traditional TF-IDF-based machine learning
- Transformer-based semantic embeddings
- I use a labeled dataset of sarcastic and non-sarcastic Reddit comments
- Goal: Determine whether combining TF-IDF with transformer embeddings leads to better sarcasm detection than only using the traditional method

Problem Statement

- How can we accurately classify sarcastic comments using machine learning models, and which feature representation, TF-IDF, or a combination of TF-IDF and transformer embeddings, yields the strongest performance?

Methodology

1. Data Preparation
   - Loaded sarcasm dataset into a pandas DataFrame.
   - Created a working copy (df_real) for analysis.
   - Cleaned text using NLTK:
   - Tokenization
   - Lowercasing
   - Stopword removal
   - Basic normalization
   - Downloaded NLTK packages (punkt, stopwords) for preprocessing.
2. Feature Engineering
   - TF-IDF Vectorization
   - Converted text into numerical vectors using TfidfVectorizer.
   - Transformer-based Embeddings
   - Generated dense semantic embeddings using a modern transformer model.
   - Combined Feature Set
   - Concatenated TF-IDF vectors + transformer embeddings to create a hybrid representation.

3. Modeling

Trained multiple classification models:

TF-IDF model (traditional machine learning classifier)

Embedding-only model

Combined fe ature model

Evaluated each using:

Accuracy

Precision

Recall

F1-score

Confusion matrices

4. Additional Unsupervised Analysis

Ran K-Means clustering on sarcastic comments.

Inspected cluster assignments to explore hidden structure in user sarcasm patterns.

	label	comment	author	subreddit	score	ups	downs	date	created_utc	parent_comment
0	0	NC and NH.	Trumpbart	politics	2	-1	-1	2016-10	2016-10-16 23:55:23	Yeah, I get that argument. At this point, I'd ...
1	0	You do know west teams play against west teams...	Shbshb906	nba	-4	-1	-1	2016-11	2016-11-01 00:24:10	The blazers and Mavericks (The wests 5 and 6 s...
2	0	They were underdogs earlier today, but since G...	Creepeth	nfl	3	3	0	2016-09	2016-09-22 21:45:37	They're favored to win.

Results

1. TF-IDF Model Performance

- Confusion Matrix:
- [[4403, 622], [3916, 1059]]
- Key Insight:
- Strong performance on non-sarcastic class (high true negatives).
- Struggles more with detecting sarcasm (lower true positives).

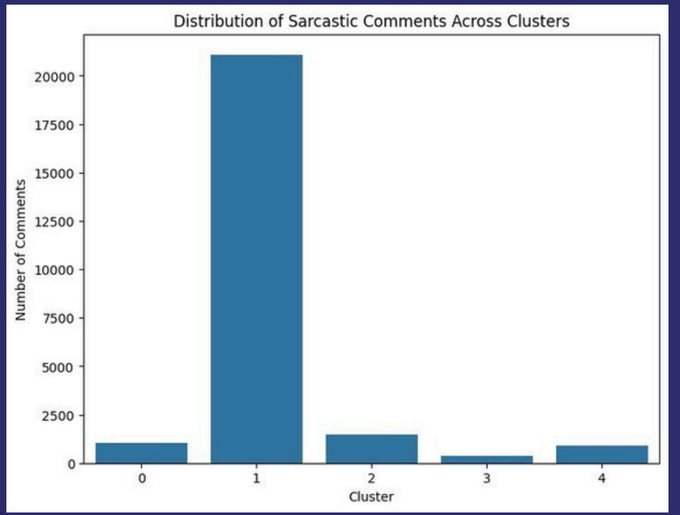
2. Combined TF-IDF + Embeddings Model

- Confusion Matrix (Combined):
- [[3302, 1723], [1951, 3024]]
- Metrics:
- Precision: 0.632
- Recall: 0.608
- F1-score: 0.622
- Interpretation:
- Best detection of sarcasm among all models.
- Balanced precision/recall → hybrid features improve nuanced detection.

```
Performance Comparison:
------------------------------
Model with TF-IDF features only:
Accuracy: 0.5462
Precision: 0.6299821534800714
Recall: 0.2128643216080402
F1-score: 0.31820913461538464
------------------------------
Model with Combined TF-IDF and Embedding Features:
Accuracy: 0.6326
Precision: 0.6370339161575732
Recall: 0.6078391959798995
F1-score: 0.622094219296441
```

3. Clustering Results

- K-Means identified distinct sub-groups of sarcastic comments.
- Indicates sarcasm varies in style/structure and can form meaningful clusters.



Future Work/Improvements

- Future improvements:
- Hyperparameter tuning
- Fine-tuned transformer model
- Explore other transformer models and embedding techniques
- Future Work:
- Explore results and clustering
- Better understand what details the model uses to determine sarcasm