Attention Isn't Enough: The Interpretability Gap in Sarcasm Detection

A0265171H, A0265159W, A0258899W, A0229920A, A0249454W
Group 30
Mentored by Yuvaraj Kumaresan
{e1053662, e1053650, e0969871, e0686119, e0941367}@u.nus.edu

Abstract

Sarcasm detection is a difficult task due to its context-dependent nature, which is a problem in news articles, as sarcastic headlines can be misinterpreted and cause misinformation. This paper focuses on sarcasm detection using headlines, investigating the trade-off between interpretability and performance. To do this, we explore 3 approaches: A model based solely on handcrafted linguistic features, a model using n-gram features, and a model that leverages attention-based neural representation. By extracting and analyzing interpretable linguistic features, replicating prior ensemble methods, and assessing the impact of augmenting attention mechanisms with context incongruity features, we demonstrate that while neural models outperform in terms of accuracy, linguistic features provide valuable interpretability, and their integration into neural networks enhances focus on semantically conflicting cues.

1 Introduction

A sarcastic statement is defined as one that is overtly untruthful and often says the opposite of what is intended (Banasik-Jemielniak et al., 2024). It is a widely used phenomenon in media, conversation, and modern culture. Whilst usually employed in relatively harmless settings (typically humorous), sarcasm in media such as journalism and news reporting is potentially harmful due to the possibility of misinformation. Given a news article, for example, it can be interpreted as its literal meaning or its exact opposite as a sarcastic statement, depending on a multitude of factors, including the source of the statement, lexical features, and a multitude of other context-based clues.

Given the context-heavy nature of sarcasm detection, as well as the potential harm of misinterpretation of sarcasm, it stands to reason that there is a need for a means of sarcasm detection. While a manual approach that involves extensive research, in-depth knowledge of the context behind the source of the statement, as well as the lexical features of sarcastic statements, could work, it would involve far too much manual labor and lacks too much generalizability to be feasible. Therefore, neural network-based or computational approaches could be explored.

In short, the goal is to be able to perform binary classification given a news article: sarcastic or genuine? Existing research shows that black-box deep learning approaches are empirically effective at producing high-accuracy models that identify sarcasm well. However, the issue with these black-box approaches is that they are black boxes. The characteristic lack of interpretability in neural embeddings means that even a high-accuracy deep learning model tells us little about the nature of sarcasm, apart from showing that it can be identified. As such, our goal and contribution to this field of research is to provide an empirical analysis on the tradeoff between interpretability through handcrafted linguistic features and the accuracy gained from black-box deep learning models.

2 Related Work / Background

A Deep neural network has been shown to perform automated sarcasm detection at an accuracy rate of around 85% without the usage of manual feature engineering (Amir et al., 2016), showing the potential of deep learning models at performing sarcasm detection binary classification. Building upon this work by using a higher-quality dataset as well as introducing a hybrid neural network approach that includes an attention span mechanism, higher accuracies nearing 90% have been achieved (Misra and Arora, 2019). The usage of bidirectional encoders from the RoBERTa transformer has also achieved F1 scores of 93% (Jayaraman et al., 2022), and incorporating contextual information into the model has improved this score to 99% (Helal et al., 2024).

On the topic of linguistic features, initial research on how lexical features contribute to sarcasm, such as interjection, punctuation, etc. has been conducted, for example, (Davidov et al., 2010) used similar lexical information to train a classifier using pattern-based lexical features to detect sarcastic Amazon product reviews. More recently, (Pradhan et al., 2024) demonstrated that handcrafted linguistic features - including readability metrics and lexical statistics can yield sarcasm detection results that are competitive with current deep learning models when combined with ensemble classifiers, achieving F1-scores of 93.75% on News headlines.

The gap in existing literature we aim to address in this paper is a lack of studies that isolate and compare linguistic features in a systematic, empirical fashion. While deep learning models have been explored, as well as linguistic features, and even some hybrid approaches that incorporate linguistic features into said deep learning models, the effectiveness and impact of the linguistic features in question have not been discussed extensively. We believe this to be a beneficial avenue to explore, as knowledge of which linguistic features contribute to defining sarcasm will greatly help in interpreting what exactly makes a statement sarcastic, a field of study that has been explored extensively.

3 Corpus Analysis and Method

The dataset used in this study is the "News Headlines Dataset for Sarcasm Detection" (Misra and Arora, 2019), which is available on Kaggle. It consists of 28,619 headlines sourced from two outlets with contrasting tones.

- Sarcastic Source: The Onion, a satire site known for its ironic and exaggerated headlines. These are labeled as sarcastic (class '1').
- Non-sarcastic Source: HuffPost, a mainstream news provider focused on factual reporting. These headlines are labeled as non-sarcastic (class '0').

3.1 Data Preprocessing and EDA

To prepare the data, we removed URLs, HTML tags, and excess whitespace and then standardized the text format. We leveraged spaCy's NLP pipeline and linguistic tools such as stopword filtering while preserving original word forms to retain linguistic nuance.

The dataset contains a rather even split: 13,634 sarcastic and 14,985 non-sarcastic headlines. This balance mitigates class bias during model training.

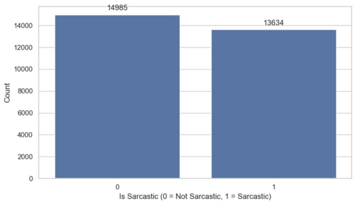


Figure 1: Distribution of Sarcastic vs Non-Sarcastic Headlines

Analysis of headline length showed a similar average length (10 words) for both classes, indicating that length alone is not a strong sarcasm indicator in this dataset. This finding motivates the need for deeper linguistic feature analysis.

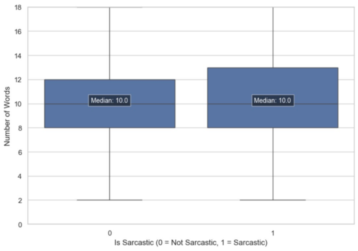


Figure 2: Headlines Length Distribution by Class

3.2 Feature Engineering

To analyze the linguistic characteristics of the headlines, we extracted a set of 30 handcrafted features from each headline. These interpretable features can be organized into four broad categories:

- **Information–theoretic & Readability** (4 features): Quantifying text complexity, lexical variety, and readability (e.g., Entropy, Lexical Diversity, Dale–Chall Score, Flesch Reading Ease).
- **Lexical & POS Counts** (9 features): Counting fundamental lexical units (e.g., Stopwords, difficult words, word lengths) and major grammatical categories from POS tags (e.g., Verb Count, Adjective Count, Noun Count).
- **Stylistic & Sentiment Markers** (9 features): Capturing explicit markers of style, tone, and affect, including counts of Negations, Intensifiers, specific punctuation, VADER-based Sentiment Score, Contrast Markers, and Average Word Length.
- **Context Incongruity** (8 features): Measuring semantic clashes or shifts within headlines, using features like Sentiment Disparity, Explicit

Incongruity counts, Positive/Negative Word counts, and sequence polarity measures.

The majority of the features (21) were based on previous research papers on sarcasm detection (Pradhan et al., 2024; Joshi et al., 2015; Liu et al., 2014), while the remaining 9 were our custom extensions based on standard sentiment-stylistic heuristics. Full definitions of all 30 features together with their origins appear in the Appendix.

4 Experiments

To examine the trade-off between interpretability and performance in sarcasm detection, we conducted three experiments using different sets of features:

1. A model based solely on the handcrafted linguistic features described in the previous section.
2. A model based on n-gram features.
3. A model using attention-based neural features.

For all experiments, we used the same data split: 80% training, 10% validation, and 10% testing, using a stratified split to preserve class distributions. The validation set was used for hyperparameter tuning, while the test set was held out for final evaluation only. The following notations are used: **Training F1** refers to the macro F1 score on the training set; **Validation F1**, on the validation set; and **Test F1**, on the testing set.

4.1 Handcrafted Linguistic Features

Our primary motivation was to determine whether we could faithfully replicate the 0.9375 F1 score reported by Pradhan et al.(Pradhan et al., 2024) using their set of 13 handcrafted linguistic features and ensemble method, that could help explain decisions made in sarcasm detection. We also explore whether augmenting that feature set could bridge any remaining performance gap.

4.1.1 Model Setup

We conducted three stages of experiments on the "News Headlines for Sarcasm Detection" dataset. We followed their preprocessing step of splitting any headline longer than ten words into multiple shorter segments before tokenization, which yielded a split of 19843 sarcastic and 21168 non-sarcastic headlines.

Initially we applied precisely the 13 features and hard-voting ensemble (decision tree, random forest, gradient boosting) described in their paper and used this as a baseline. Then, we repeated the same pipeline with 2 modifications mentioned below to see if the F1 score improved:

- Adding 17 additional sentiment–stylistic features (for a total of 30 as mentioned in section 3.2):
- Removing the Pre-processing step of splitting headlines

Following the paper, we used a single 80/20 stratified split, holding out 20% of the data as our test set and training exclusively on the remaining 80. Random Grid Search was used to fine-tune the parameters of the full ensemble.

4.1.2 Combined Results

Table 1 reports the test-set macro-average F1 for each pipeline.

Model	**Train F1**	**Test F1**
Replica Ensemble with 13 features	0.77	0.66
13 base features + 17 extended features	0.8	0.67
- headline splitting in preprocessing	0.87	**0.71**

Table 1: Test-set macro-F1 for models using only linguistic features.

4.1.3 Feature Importance

To identify which features drive the ensemble's predictions, we perform a permutation-importance analysis: we shuffle each feature in turn on the test set and measure the drop in macro-F1.

Feature	$\Delta F1$
Verb Count	0.0678
Stop word Count	0.0509
Single-letter Word Count	0.0456
Nouns	0.0309
Adjectives	0.0113
Wrong words	0.0100

Table 2: Top five most important features based on permutation importance on the ensemble model

Purely linguistic features have performance ceilings, motivating our next experiments using n-gram context and attention mechanisms.

4.2 N-gram Features

N-grams can capture word frequencies and sequential patterns in a document, which may contribute to the detection of sarcasm by identifying exaggerated word choices and recurrent word pairings (e.g., bigrams) that are statistically common and potentially indicative of sarcastic expressions.

4.2.1 Model Setup

This experiment trains models on n-gram features and aims to identify useful features. The process begins with selecting an ML algorithm. Three algorithms were evaluated on unigram features extracted using `CountVectorizer`: Multinomial Naive Bayes (NB), Logistic Regression (LR), and a Neural Network (NN).

The hyperparameters used include `max_iter=150` for LR to ensure convergence, and a hidden layer of size (15,) with `max_iter=70`, and `early_stopping=True` for NN to reduce training time. Although LR had the lowest validation F1 score (0.8429), the scores for NB and NN were similar (NB: 0.8470, NN: 0.8444). LR was ultimately chosen for its interpretability, as it is a linear model and allows direct analysis of feature weights.

Next, the n-gram range is tuned. After testing different ranges, `ngram_range=(1, 2)` gives the highest validation F1. It outperforms `ngram_range=(1, 1)` because bigrams can capture more context. One consideration is that headlines are often just one or two sentences, so using larger n-gram sizes might risk overfitting.

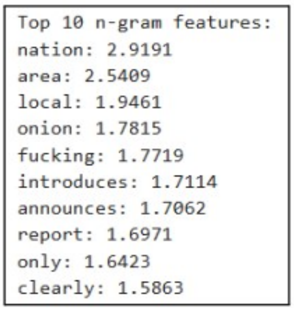
Attempts to remove stopwords led to a slight drop in validation F1 by approximately 0.05. Given that the training F1 score is higher than the validation F1 by about 0.14, the regularization parameter C in Logistic Regression is reduced to improve generalization. After experimenting, C=0.5 is selected as the optimal value. Our model has **Test F1 of 0.8495**.

Model Variant	Description	Train F1	Val F1
Baseline	Unigram features + LR	0.9609	0.8429
N-gram Expansion	Unigram + bigram features with LR	0.9989	0.8544
Lower C	LR with C=0.5 (default value is 1)	0.9943	0.8545

Table 3: Comparison of N-gram Model Variants

4.2.2 Feature Importance

The top 10 features with the highest positive weights in the LR model are all unigrams. The fact that only unigrams appear among the top features suggests that the model places significant emphasis on word choice when detecting sarcasm. **Figure 3** shows these features along with their corresponding weights.



```
Top 10 n-gram features:
nation: 2.9191
area: 2.5409
local: 1.9461
onion: 1.7815
fucking: 1.7719
introduces: 1.7114
announces: 1.7062
report: 1.6971
only: 1.6423
clearly: 1.5863
```

Figure 3: Top 10 N-gram Features by Weight

4.3 Attention-Based Model

To investigate the contribution of handcrafted linguistic features—particularly explicit context incongruity—on sarcasm detection, we conducted an empirical study building upon prior work by Misra and Arora (Misra and Arora, 2019). Their architecture uses a hybrid neural network consisting of a CNN-LSTM-attention pipeline to classify news headlines from our dataset into binary labels: sarcastic or non-sarcastic. Their attention-based framework already demonstrates state-of-the-art performance on this dataset, making it a suitable baseline for comparison.

Our goal is to examine whether integrating handcrafted linguistic features, specifically explicit incongruity, can improve the model's ability to locate and attend to sarcasm-indicative cues. This stems from linguistic literature that defines sarcasm as a function of failed expectation or semantic conflict (Campbell and Katz, 2012). Such incongruity often manifests when a strongly positive expression is juxtaposed with an evidently negative scenario, such as *"Being stranded in traffic is a great start to the day"*. Joshi et al. (Joshi et al., 2015) previously categorised these incongruities into *explicit* (clearly divergent sentiments in adjacent tokens) and *implicit* (where contrast relies on background world knowledge). We focus on explicit incongruity as it is more detectable within the confines of short-form text like headlines.

4.3.1 Feature Design

To quantify explicit incongruity, we introduce a feature extraction pipeline based on sentiment polarity transitions. Using the VADER `SentimentIntensityAnalyzer`, each word in a headline is assigned a compound sentiment score. Polarity transitions (positive to negative or vice versa) are counted as instances of incongruity. We also extract auxiliary statistics such as the number of positive/negative words and the longest contiguous subsequences of same-polarity tokens. This feature set captures both local sentiment discontinuities and lexical polarity trends—which we hypothesise are strong indicators of sarcasm in formal headlines.

4.3.2 Experimental Setup

We compare two models:

- **Baseline Model:** A CNN-LSTM-attention model trained solely on token embeddings.
- **Enhanced Model:** The same architecture augmented with the explicit incongruity feature concatenated into the input vector before the attention module.

Both models are trained on the same dataset using identical hyperparameters. During training, we monitor the training and test F1-score and attention visualisations to assess performance and interpretability.

4.3.3 Results and Analysis

Figure 4 displays side-by-side attention heatmaps for selected headlines under both the baseline and enhanced models. Across multiple examples, the enhanced model demonstrates more focused and semantically coherent attention distributions. For instance, in the headline *"Wheelchair bound student would have preferred to sit out pep rally"*, the enhanced model sharpens its focus on *"preferred"*, *"sit out"*, and *"pep rally"*, a juxtaposition that typifies sarcastic tone due to clashing expectations. This contrasts with the baseline model, which dilutes attention across less relevant tokens.

To assess model performance over time, we tracked training and test F1 scores across epochs for both the baseline and enhanced models. **Figures 5** and **6** present these trends side by side.

The baseline model demonstrates strong learning on the training data, with F1 scores rising from 0.81 to nearly 0.98. However, its test F1 score shows a slight but consistent decline, dropping from 0.86 to 0.84, a sign of overfitting. In contrast, the enhanced model achieves a similarly high training F1 but shows more variation and resilience in its test scores. After an initial dip, its validation F1 score begins to recover by epoch five, ending slightly higher than the baseline.

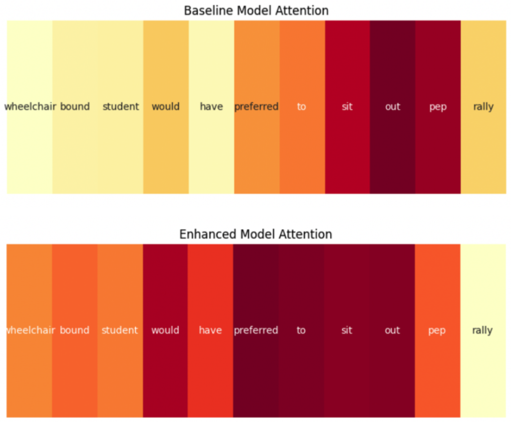


Figure 4: Comparison of attention weights between the baseline and enhanced models for sample headlines. The enhanced model shows improved focus on semantically conflicting phrases.

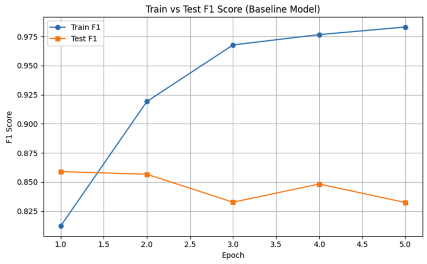


Figure 5: Baseline model F1 score over 5 epochs. Test performance decreases slightly despite rising training accuracy, indicating overfitting.

The enhanced model's relative stability in test performance suggests that incorporating handcrafted incongruity features may help mitigate overfitting and improve generalisation. Although the performance improvement is modest, this aligns with the model's theoretical motivation of enhancing interpretability and enabling the attention mechanism to better identify contextually clashing phrases, which are often characteristic of sarcasm. This indicates that linguistic features such as explicit incongruity, while not drastically boosting accuracy, provide added robustness and insight into model decision-making.

5 Discussion

5.1 Experiment 4.1 Discussion

Why are we unable to replicate the F1=0.9375 reported by Pradhan et al. (2024) when applying their 13-feature ensemble method?

Despite following Pradhan et al.'s preprocessing steps, feature set (13 handcrafted features), and

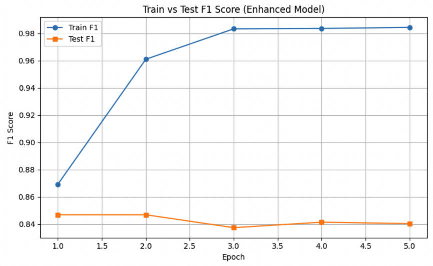


Figure 6: Enhanced model F1 score over 5 epochs. Test performance is more stable and recovers by the final epoch.

hard-voting ensemble (decision tree + random forest + gradient boosting), our baseline plateaus at an F1 of 0.66 on the test split by over 25 points below their reported 0.9375. A modest gain to 0.67 arises when we augment to 30 features, and removing the headline-splitting step recovers only F1 = 0.71. Several factors likely conspire to explain this gap:

- As they did not mention how exactly they split the headlines during preprocessing, they may have implemented more sophisticated sentence-boundary rules that we could not fully reproduce. Our simpler chunking may have fragmented context in ways that hurt the ensemble's ability to learn the intended patterns.

- We used a Random Grid Search over roughly the same ranges, but the paper does not specify the exact hyperparameter values for each base learner. The lightly tuned parameters in our runs may underfit or overfit relative to their finely-tuned settings.

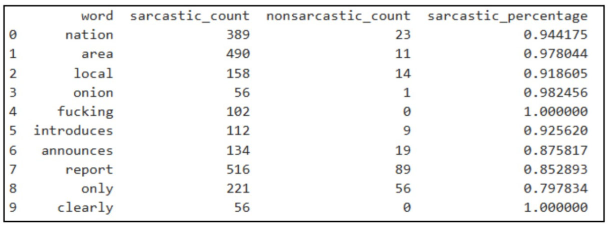
- Our permutation analysis shows that the top-five most critical features (Counts of verbs, stop words, single words, nouns, adjectives and wrong words) all originate in the original 13-feature set. The 17 additional features we introduced yielded only marginal gains (0.01 in F1), indicating that the core "handcrafted" signals have been fully captured, and yet still fall short of 0.93. This suggests that no simple extension of interpretable features, nor headline splitting, can recover the published result.

Taken together, these observations imply that the 0.9375 F1 in Pradhan et al. may rely on unpublished details such as data splits and ensemble configurations that are not fully disclosed.

5.2 Experiment 4.2 Discussion

Are the n-gram features truly indicative of sarcasm, or do they reflect stylistic tendencies of specific sources (e.g., author habits or editorial styles of particular websites)?

Figure 7 shows the frequency of the top 10 n-gram features in sarcastic versus non-sarcastic headlines. The "sarcastic percentage" refers to the proportion of each word's occurrences in sarcastic headlines relative to its total occurrences across all headlines.



```
          word  sarcastic_count  nonsarcastic_count  sarcastic_percentage
0       nation              389                  23              0.944175
1         area              490                  11              0.978044
2        local              158                  14              0.918605
3        onion               56                   1              0.982456
4      fucking              102                   0              1.000000
5   introduces              112                   9              0.925620
6    announces              134                  19              0.875817
7       report              516                  89              0.852893
8         only              221                  56              0.797834
9      clearly               56                   0              1.000000
```

Figure 7: Frequency and Sarcastic Percentage of Top 10 N-gram Features

The top 10 features are more likely to appear in sarcastic headlines. Among these words, *"clearly"* and *"fucking"* appear exclusively in sarcastic headlines. In the context of headlines, these words can serve as indicators of sarcasm. Genuine headlines generally prioritize objectivity and factual accuracy, avoiding unnecessary emphasis or emotional language.

In non-sarcastic headlines, the word *"clearly"* is rarely used because it may be seen as redundant when stating facts. It also risks assuming that the information is self-evident to all readers. In contrast, sarcastic headlines often employ *"clearly"* to frame subjective or unproven claims with irony or humor. For example, *"Olympics officials clearly trying to buy more time with a 6-day-long opening ceremony performance."*

In sarcastic headlines, the word *"fucking"* functions as an intensifier, adding exaggerated emotion or emphasis to the statement. For example: *"Terrible fucking taste sweeps Teen Choice Awards."* This kind of language can create a sharp contrast, often amplifying the irony. However, it's important to note that this word is not an ideal feature, as news providers like *HuffPost* would avoid such language due to its offensive and vulgar nature. It could be seen more as a stylistic choice of certain websites, rather than an indication of sarcasm.

The word *"onion"* ranks as the 4th highest-weighted feature. Since there is not much serious news about onions, sarcastic headlines are more likely to mention the word *"onion."* However, the high weight here is more likely due to *The Onion* publishing self-referential headlines, such as those involving its CEO, staff reporters, or social media users. This feature reflects *The Onion*'s tendency to reference itself, rather than being a general sign of sarcasm.

We can observe that some features indicate sarcasm in the context of headlines, while others reflect the stylistic tendencies of specific sources. Using n-gram features with linear models like LR provides interpretable results. However, using word choice as an indicator of sarcasm may be difficult to generalize to other contexts. For example, some words that rarely appear in non-sarcastic headlines might still occur in non-sarcastic texts on social media.

5.3 Experiment 4.3 Discussion

Can handcrafted linguistic features enhance the interpretability and performance of attention-based neural models for sarcasm detection?

To better understand how sentiment-based linguistic features interact with attention mechanisms in sarcasm detection, we conducted a fine-grained attention difference analysis. Specifically, we measured whether the sentiment polarity of individual words (as computed using the VADER `SentimentIntensityAnalyzer`) correlates with their change in attention weight after the inclusion of handcrafted incongruity features.

Figure 8 shows attention heatmaps for the sentence *"nation excited for some insane k-pop shit during opening ceremony"* under both the baseline and enhanced models. Figure 9 visualises the relationship between sentiment score and change in attention weight (denoted ΔAttention = Enhanced - Baseline) for each token. A Pearson correlation coefficient was computed to evaluate the strength of this relationship.

Key Observations.

- **Correlation coefficient = -0.03:** This weak negative correlation suggests that sentiment magnitude does not meaningfully drive changes in attention. Words with highly positive or negative sentiment were not consistently emphasised more after introducing incongruity features.

- **Largest attention differences not sentiment-driven:** Words such as *'some'* (neutral), *'excited'* (positive), and *'insane'* (negative) demonstrated the highest ΔAttention values. This indicates that the attention changes were likely influenced by *contextual contrast* or *semantic incongruity*, rather than raw sentiment polarity.

- **Example: *'excited'* and *'insane'*:** Although *'excited'* is a positive word, its juxtaposition with *'insane'* and *'shit'* creates a tone of exaggeration or irony that is typical of sarcasm. The enhanced model's ability to adjust attention weights toward these interactions reflects an improved grasp of such linguistic subtleties.

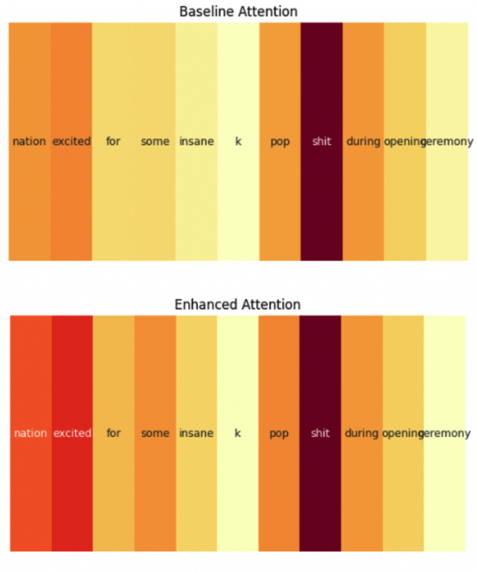


Figure 8: Attention heatmaps from the baseline (top) and enhanced (bottom) models for the same input sentence. The enhanced model shows improved emphasis on emotionally and contextually salient terms.

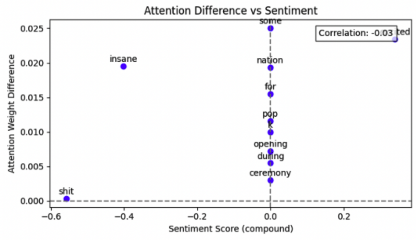


Figure 9: Scatterplot showing the relationship between sentiment score and attention weight difference for each token. Correlation coefficient = -0.03.

These findings reinforce our hypothesis that *context incongruity*, rather than standalone sentiment, is a stronger cue for sarcasm. By incorporating handcrafted features such as polarity shifts and incongruity counts, the model better detects conflicting emotional tones across adjacent words. While it might be assumed that emotionally charged words naturally receive more attention, our analysis demonstrates that it is the *interaction* of sentiments (e.g., a positive expression following a negative premise) that most reliably signals sarcasm.

This aligns with the theoretical framing offered by Campbell and Katz (Campbell and Katz, 2012) and Joshi et al. (Joshi et al., 2015), who emphasise sarcasm as a form of expectation violation or semantic dissonance. The enhanced model's behaviour supports this view by focusing attention not merely on strong affective tokens, but on those that clash or subvert expectations within a given context.

6 Conclusion

The three experiments highlight the trade-off between interpretability and performance in sarcasm detection.

Experiment 4.1: The ensemble method with 30 linguistic features is fully transparent, as feature weights map directly to human-readable cues. However, its macro-F1 plateaus around 0.71, showing that maximal interpretability alone cannot match the accuracy of models using n-grams and attention.

Experiment 4.2: The linear model (LR) using n-gram features achieves a test F1 score of around 0.85, indicating strong performance. Moreover, examining the features with the highest weights offers some interpretability. The top 10 weighted features are words that occur more frequently in sarcastic headlines from *The Onion*.

Experiment 4.3: This experiment investigated a hybrid approach that integrates handcrafted incongruity features into a neural attention-based model. While the raw F1 performance gains were modest, the enhanced model exhibited improved generalisation and clearer attention patterns over sarcasm-indicative tokens. Attention visualisations and sentiment interaction analyses revealed that the model could better capture contextual dissonance. This aligns with theoretical definitions of sarcasm as a violation of expectation.

Limitations. Our permutation analysis in section 4.1 revealed that seemingly salient cues such as *verb count*, *stop-word count*, and *one-letter words* derive largely from stylistic quirks of *The Onion* rather than any inherent property of sarcasm. Consequently, the reported gains may not transfer to other datasets, underscoring the need for cross-source evaluation.

While this study utilizes the "News Headlines Dataset for Sarcasm Detection," certain features identified in the experiment may not generalize well to other contexts. For example, the high weights of n-gram features are likely influenced by the specific composition of the dataset, which includes headlines exclusively from *The Onion* and *HuffPost*. These features may not be applicable to other contexts, such as social media platforms or headlines from other news sites, where language use, tone, and contextual cues may differ significantly.

Future Direction. Future work should examine whether the benefits of feature–attention integration hold in more diverse and noisy text settings, such as Twitter or Reddit. Beyond optimizing for performance, future models should also emphasize interpretability, ensuring that the results are not only accurate but also understandable and actionable in real-world applications.

References

Silvio Amir, Byron C. Wallace, Hao Lyu, Paula Carvalho, and Mário J. Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. *CoRR*, abs/1607.00976.

Natalia Banasik-Jemielniak, Piotr Kałowski, and Maria Zajaczkowska. 2024. *Studying Verbal Irony and Sarcasm: Methodological Perspectives from Communication Studies and Beyond*, volume 1. Springer Nature Switzerland.

John D Campbell and Albert N Katz. 2012. Are there necessary conditions for inducing a sense of sarcastic irony? *Discourse Processes*, 49(6):459–480.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcasm in Twitter and Amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 107–116, Uppsala, Sweden. Association for Computational Linguistics.

Nivin Helal, Ahmed Abdelgawad, Nagwa Badr, and Yasmine Afify. 2024. A contextual-based approach for sarcasm detection. *Scientific Reports*, 14.

Ashok Kumar Jayaraman, Tina Trueman, Gayathri Ananthakrishnan, Satanik Mitra, Qian Liu, and Erik Cambria. 2022. Sarcasm detection in news headlines using supervised learning. pages 288–294.

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 757–762, Beijing, China. Association for Computational Linguistics.

Peng Liu, Weiming Chen, Guangwei Ou, Tian Wang, Dong Yang, and Ke Lei. 2014. Sarcasm detection in social media based on imbalanced classification. In Vincent S. Tseng, Tu Bao Ho, Zhi-Hua Zhou, Arbee L. P. Chen, and Hung-Yu Kao, editors, *Advances in Knowledge Discovery and Data Mining*, volume 8443 of *Lecture Notes in Computer Science*, pages 459–471. Springer.

Rishabh Misra and Prahal Arora. 2019. Sarcasm detection using hybrid neural network. *CoRR*, abs/1908.07414.

Jitesh Pradhan, Rajshree Verma, Sumit Kumar, and Varun Sharma. 2024. An efficient sarcasm detection using linguistic features and ensemble machine learning. *Procedia Computer Science*, 235:1058–1067. International Conference on Machine Learning and Data Engineering (ICMLDE 2023).

Acknowledgements

We would like to express our sincere thanks to Yuvaraj Kumaresan for supervising our project and providing insightful advice, as well as Prof. Kan Min Yen and the NUS CS4248 teaching team for their guidance and assessment of the project.

Statement of Independent Work

1A. Declaration of Original Work. By entering our Student IDs below, we certify that we completed our assignment independently of all others (except where sanctioned during in-class sessions), obeying the class policy outlined in the introductory lecture. In particular, we are allowed to discuss the problems and solutions in this assignment, but have waited at least 30 minutes by doing other activities unrelated to class before attempting to complete or modify our answers as per the class policy.

This particular document did not use any AI Tools to proofcheck and was constructed and edited purely by manual work.

Signed,
A0265171H, A0265159W, A0258899W, A0229920A, A0249454W

e1053662, e1053650, e0969871, e0686119, e0941367
@u.nus.edu

A Appendix

Full Linguistic Feature Definitions

#	**Feature Name**	**Description**
Features 1–13 (Pradhan et al., *2024)*		
1	Entropy	Shannon entropy of the token frequency distribution.
2	Lexical Diversity	Unique token count divided by total tokens.
3	Dale–Chall Score	Readability index based on difficult words & sentence length.
4	Flesch Reading Ease	Readability score (sentence length & syllable count).
5	Stopword Count	Number of stop-words (e.g. “the”, “is”).
6	Wrong Word Count	Tokens absent from standard lexicon.
7	Difficult Word Count	Tokens not in Dale–Chall easy list.
8	Lengthy Word Count	Words of length > 2 characters.
9	Two-Letter Word Count	Words of exactly 2 letters.
10	One-Letter Word Count	Single-letter words.
11	Verb Count	Number of spaCy-tagged verbs.
12	Adjective Count	Number of spaCy-tagged adjectives.
13	Noun Count	Number of spaCy-tagged nouns.
Custom Features (this work)		
14	Negation Count	Count of negation words (e.g. “not”, “never”).
15	Intensifier Count	Count of words like “very”, “really”.
16	Downtoner Count	Count of words like “slightly”, “somewhat”.
17	Exclamation Mark Count	Number of ‘!’ characters.
18	Question Mark Count	Number of ‘?’ characters.
19	Sentiment Score	VADER compound sentiment of the headline.
20	Contrast Marker Count	Count of “but”, “however”, etc.
21	Average Word Length	Mean character length of alphabetic words.
22	Sentiment Disparity	Mean weighted polarity difference across token pairs.

Table 4: Linguistic features 1 - 22

#	**Feature Name**	**Description**
Features 23–29 (Joshi et al., *2015)*		
23	Explicit Incongruity Count	Count of adjacent tokens with opposite polarity.
24	Largest Positive Subsequence	Longest run of positive-polarity tokens.
25	Largest Negative Subsequence	Longest run of negative-polarity tokens.
26	Largest Same Polarity Sequence	Longest run of same non-neutral polarity.
27	Positive Word Count	Number of tokens with polarity > 0.05.
28	Negative Word Count	Number of tokens with polarity < –0.05.
29	Lexical Polarity	Sum of all token-level polarity scores.
Feature 30 (Liu et al., *2014)*		
30	Semantic Imbalance Rate (SIR)	Avg. of each word’s max cosine similarity to any other word (lower = more spread).

Table 5: Linguistic features 23 - 30