Sarcasm Detection

Advanced Computational Learning and Data Analysis- 52025

Final Project

Paper, Github: [1],[2]

Submitted by: Itay Matityahu, Daniel Nissani
To: Yuval Benjamini, Ibrahim Bashir

האוניברסיטה העברית בירושלים
THE HEBREW UNIVERSITY OF JERUSALEM

Background

Our work is based on the paper *"Sarcasm Detection using Hybrid Neural Network"* (Rishabh Misra and Prahal Arora, 2019). Sarcasm detection is considered an especially challenging task for both humans and machine learning models, as it requires understanding irony and contextual meaning beyond the literal interpretation of words. Accurate sarcasm detection is important for many downstream NLP applications, such as sentiment analysis, opinion mining, and content moderation, where misinterpreting sarcastic expressions may lead to incorrect conclusions.

In their work, the authors focus on sarcasm detection in news headlines, a particularly challenging setting due to the short length of the text and the limited contextual information. The dataset consists of two types of headlines: real news headlines from *HuffPost* and sarcastic headlines from *The Onion*. The dataset is relatively balanced, containing 14,984 real headlines and 11,725 sarcastic ones.

The proposed model consists of three main components: Word2Vec embeddings, a CNN layer, and a BiLSTM with an attention mechanism. This architecture is well suited for the task, as it captures complementary linguistic signals present in sarcastic text. The CNN component focuses on local n-gram patterns and salient word-level cues, while the BiLSTM models sequential dependencies by incorporating both past and future context within the sentence. The attention mechanism further allows the model to emphasize informative tokens that are critical for identifying sarcastic intent in short headlines.

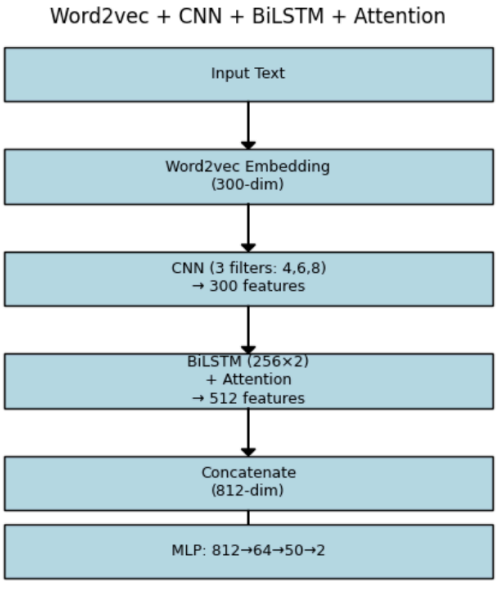
Method and Modifications

Comparative Study of Architectures:

In this study, we evaluate the performance of a baseline hybrid model against two proposed architectures. Our primary objective is to investigate the transition from static embeddings (Word2Vec) and manual feature engineering to dynamic, transformer-based self-attention mechanisms (DEBERTA).

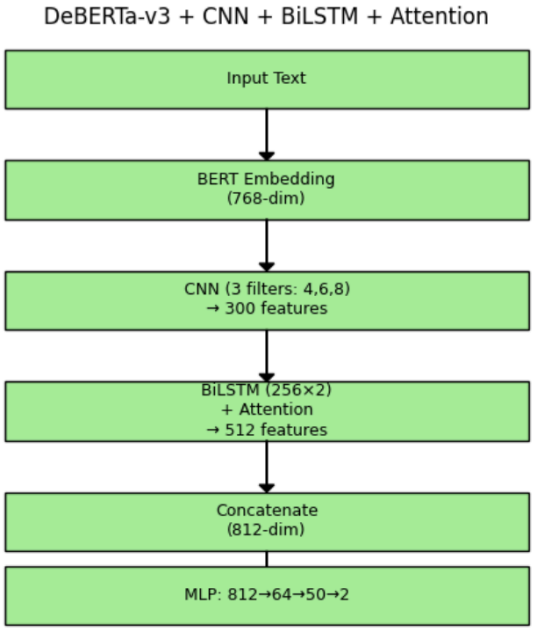
Baseline Model: Word2Vec + CNN & BiLSTM

Following the methodology of Misra & Arora (2019), we reconstructed the baseline model utilizes a parallel hybrid structure. Headlines are converted into Word2Vec embeddings, which serve as input to both a CNN and a BiLSTM and attention branch simultaneously. The CNN extracts local features, while the BiLSTM use attention to captures sequential dependencies. The outputs are concatenated and fed into an MLP for the final prediction.



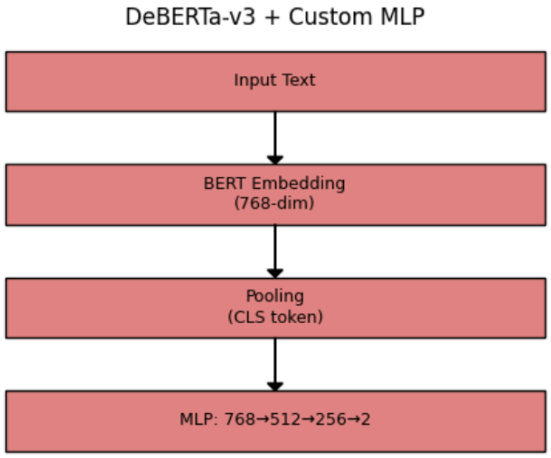
Model 1: DEBERTA + CNN & BiLSTM

Our first proposed model retains the exact hybrid architecture of the baseline but replaces Word2Vec with DEBERTA embeddings. This specific configuration was designed to isolate the contribution of DEBERTA to the overall performance. By keeping the rest of the architecture identical, we can directly measure how much the DEBERTA embedding improves (or not) the feature extraction process compared to static embeddings within the same framework. DEBERTA is a contextual transformer-based embedding model that disentangles content and positional information in its attention mechanism, enabling richer, context-dependent word representations than static embeddings such as Word2Vec.



Model 2: DEBERTA + Custom Head

The second proposed model shifts toward a more streamlined architecture, consisting of DEBERTA followed by a compact MLP. We hypothesize that this will be the most powerful model in our study. Because DEBERTA inherently utilizes multi-head self-attention to weigh word importance and context, we eliminate the need for manual attention layers. This approach relies on the transformer's internal ability to represent complex sarcastic patterns directly.



Architectural Evolution

The primary motivation for transitioning from the hybrid CNN-BiLSTM structure to a standalone DEBERTA model was to leverage the significant advancements in natural language understanding that emerged after the baseline paper's publication.

While the original architecture relied on manual attention modules and parallel layers to capture context, DEBERTA natively integrates these dependencies through its self-attention mechanism.

We chose DEBERTA specifically because it consistently outperforms other models on natural language understanding tasks by better modeling the relationship between word pairs, which is crucial for identifying the "clash" of meanings often found in sarcastic headlines.

Optimization and Loss Function

To ensure a fair comparison and focus strictly on the architectural performance, we utilized the same loss function as the baseline paper: Binary Cross-Entropy (BCE). This function is particularly suited for this mission because sarcasm detection is a binary classification task at its core.

BCE effectively measures the divergence between the predicted probability of sarcasm and the actual label which is essential when dealing with the thin boundary between sarcastic and non-sarcastic text.

Experiments and Analysis

Experiments methodology and Accuracy measure

The experimental part of our work involved training and evaluating the three architectures described above. The dataset was divided into training, validation, and test sets. The validation set was used for model selection only for the final architecture (BERT with a custom classification head), while for the other two architectures we adopted the hyperparameters reported in the original paper.

To compare the performance of the three models, we used classification accuracy as the primary evaluation metric. We chose accuracy in order to remain consistent with the metric used by the authors of the original work, allowing for a direct and fair comparison of performance improvements. Moreover, accuracy is an appropriate metric in this setting because the dataset is relatively balanced, and there is no asymmetric cost associated with false positives or false negatives. Unlike domains such as medical diagnosis, where minimizing false negatives is critical, or fraud detection, where false positives may be more tolerable, sarcasm detection does not impose a strong preference toward either type of error.

Our methodology for comparing the models was designed to isolate and evaluate the contribution of different architectural components in the original model. By modifying one component at a time, we aimed to understand which changes were responsible for performance improvements. Consequently, all architectural modifications were introduced gradually.

In the first stage, we trained the original model proposed in the paper in order to reproduce the reported results and establish a reliable baseline. This step allowed us to verify the correctness of the implementation and to ensure that subsequent comparisons were grounded in a faithful reproduction of the original architecture.

In the second stage, we modified only the encoder component of the original model by replacing the Word2Vec embeddings with a modern contextual encoder, DEBERTA, while

preserving the remaining architecture, including the CNN and BiLSTM components. The model was trained on the same dataset using the same hyperparameters as the original implementation. This modification was expected to improve the quality of the textual representations and better capture contextual relationships between words.

In the third stage, we retained the modern DEBERTA encoder and removed the original CNN and BiLSTM layers and then passed it through a simpler multi-layer perceptron (MLP) classification head. This change was motivated by the observation that transformer-based encoders inherently model long range dependencies through self attention mechanisms, potentially eliminating the need for recurrent architectures such as BiLSTMs.

Fine Tuning

For the final model (DEBERTA + custom head), which serves as our primary architecture, we performed a grid search as part of the fine-tuning and model selection process. Specifically, we evaluated multiple hyperparameter configurations, including the learning rate, pooling strategy (CLS or mean pooling), learning rate scheduler type, and the depth of the classification head, corresponding to the number of layers used for feature compression prior to classification. We test all the optional models on the validation set so the test set will be used only for evaluating the final model.

Uncertainty Estimation in Model Comparison

In order to quantify the uncertainty of the evaluated models and assess the stability of their performance across different samples, we applied a bootstrap based evaluation procedure. Bootstrap resampling allows us to approximate the sampling distribution of a performance metric without assuming a specific parametric form. Specifically, we repeatedly sampled, with replacement, from the test set to create multiple bootstrap test sets of the same size as the original test set.

For each bootstrap sample, we generated predictions using the fixed, fully trained model and computed the classification accuracy. This process was repeated 100 times, resulting in a bootstrap distribution of accuracy values. Based on this distribution, we estimated the mean accuracy and constructed a non-parametric confidence interval (CI), which provides an empirical measure of the variability of the model's performance due to test set sampling.

It is important to note that this bootstrap procedure captures uncertainty arising from the finite size of the test set, while keeping the model parameters fixed. As such, it reflects the robustness of the reported performance with respect to variations in the evaluation data, rather than uncertainty induced by the training process itself.

Results and Discussion

MODEL	Test Accuracy	CI (95%)
Baseline - Word2Vec + CNN & BiLSTM	88%	88 ± 0.64
DEBERTA + CNN & BiLSTM	86.2%	86.2 ± 0.58
DEBERTA + Custom Head	**94.4%**	**94.4 ± 0.13**

The results of our comparative study demonstrate a clear distinction between traditional hybrid architectures and modern transformer-based approaches. As shown in our evaluation, DEBERTA + Custom Head emerged as the superior model, achieving a 94.4% test accuracy with a very tight 95% Confidence Interval of ±0.13.

Interestingly, the hybrid model using DEBERTA embeddings (86.2%) slightly underperformed compared to the original Word2Vec baseline (88%). This suggests that simply swapping static embeddings for contextual ones within a manually designed CNN-BiLSTM framework does not necessarily yield better results, maybe because those architectural layers are optimized for different feature distributions than what DEBERTA provides.

The significant leap in performance seen in the DEBERTA + Custom Head model can be attributed to the inherent power of the transformer's self-attention mechanism. While the baseline and the hybrid model attempt to manually capture local and global context through CNN and BiLSTM layers, DEBERTA performs these operations more effectively within its internal layers.

Examples

Headline	True Label	Predicted Label	Accurate
substitute teacher can tell he's filling in for real asshole	**sar**	**sar**	**Yes**
at least 16 dead after train crashes into bus in Mexico	**non**	**non**	**Yes**
world's youngest person born	**sar**	**non**	**No**
pregnant Kourtney Kardashian rocks all black everything	**non**	**sar**	**No**