LOLgorithm: Integrating Semantic,Syntactic and Contextual Elements for Humor Classification

Tanisha Khurana
Electrical Engineering
tkhuran3@ncsu.edu

Vikram Pande
Electrical Engineering
vspande@ncsu.edu

Kaushik Pillalamarri
Computer Science
spillal2@ncsu.edu

I. Problem Description

Humor is a fascinating and puzzling area of study in the field of computers understanding human language. When a computer talks to a person, if it can understand and appreciate the humor in what the person is saying, it can grasp the real meaning of human language better. This, in turn, helps the computer make better decisions to enhance the user's experience. So, working on techniques that allow computers to "get" humor in human conversations and adjust their responses accordingly is something we should pay special attention to.

Humor Recognition is about figuring out if a sentence is funny in a given situation. It's a tough problem in understanding human language. First, humor is tricky because different people may find different things funny even in the same sentence. Second, humor often depends on knowing a lot about the context. For example, think about these sentences: "The one who invented the door knocker got a No Bell prize" and "Veni, Vidi, Visa: I came, I saw, I did a little shopping." To find them funny, you need to know a bit about culture and language. Lastly, there are many types of humor, like wordplay, irony, and sarcasm, but we don't have a clear way to categorize them all. So, it's almost impossible to create a computer program that can recognize all types of humor, just like humans can't always classify them perfectly either.

In this work, we formulate humor recognition as a classification task in which we distinguish between humorous and non-humorous instances. Exploring the syntactical structure involves leveraging Lexicons to capture sentiment counts within a sentence, while Statistics of Structural Elements (SSE) encapsulates the statistical insights of Noun phrases, Word phrases, and more. Unveiling the semantic layers of humor delves into Word2Vec embeddings, analyzing incongruity, ambiguity, and phonetic structures within sentences. Additionally, contextual information is harnessed through ColBERT embeddings. For each latent structure, we design a set of features to capture the potential indicators of humor.

A. Example Scenario

Imagine you work for a social media analytics company, and one of your clients is interested in understanding and quantifying humor within user-generated content on their platform. They want to identify and categorize humorous posts, as well as analyze the types of humor that resonate the most with their audience.

Amazon for example employs humor detection in product question-answering systems[2] as some products attract humor due to their unreasonable price and their peculiar functionality. Detecting humorous questions in such systems is important for sellers, to better understand user engagement with their products. It is also important to inform users about the flippancy of humorous questions, and that answers for such questions should be taken with a grain of salt.

II. DATASET

We employ two distinct datasets, for training and testing purposes:

ColBERT Dataset: The ColBERT dataset has 200,000 statements with humor labels and is accessible on Kaggle for model training and evaluation. We partition it into training and validation sets.

For the unseen test set, we scraped funny one-liners from the Bestlife website to gather jokes. We preprocess the data by removing punctuation and lemmas and manually labeling the jokes as humorous or not for unseen datasets for model evaluation. Also we've obtained around 2K samples of sentences from reddit, out of which some are funny are and some are not, which would also be used for further evaluating the model's performance on unseen data.

A. CODE

We use libraries such as PyTorch, TensorFlow, HuggingFace, NLTK, SpaCy, NRCLex, Sklearn, and some inherent Python libraries to help in the coding part.

III. Problem importance

Humor classification is a crucial problem and our solution takes into account the contextual, semantic, and syntactic meaning of the joke. Our problem has various applications in customer support and chatbot development. Many companies utilize conversational services to recommend items and counsel consumers. There is also a growing demand for emotional chatterbots for users' higher satisfaction in various commercial fields. Siri could become more 'human' if she had the ability to recognize social cues like humor and respond to them with laughter. As described in our example scenario, our problem will be useful in social media analytics and for question-answering systems for products on Amazon[2]. Our problem also has further usage when combined with speech as well. For

example, sitcoms are written and performed primarily to cause laughter, so the ability to detect precisely why something will draw laughter is of extreme importance to screenwriters and actors[5].

IV. Methodology

The aim of the project was to work more with linguistic features rather than exploring computational methods in the area of Natural Language Processing. For the features we chose to work with we've categorized them into three types for experimentation: syntactic, semantic, and contextual features. Syntactic features help understand how words function structurally and influence model predictions. Semantic analysis delves into word and sentence meanings and contextual dives into the context of the joke. We're generating these features using methods described later in this section and analyzing these features to identify the most influential ones driving model predictions using basic SHAP interpretations, and visualizing decision tree.

A. *Syntactical information*

1) ***Lexicons****:* In some cases, individual words can be inherently funny. Therefore, if a sentence contains amusing words, the entire sentence may be humorous. With this assumption in mind, our initial approach involves extracting syntactical information using lexicons [1],[13]. Specifically, we employ the NRC word emotion lexicons for this purpose. Utilizing this lexicon, we represent words as vectors, where each entry signifies the word's similarity to various emotions such as anger, anticipation, disgust, fear, joy, sadness, and surprise, as well as its similarity to the general sentiments of positivity and negativity. By employing these word representations, we aim to capture the syntactical information embedded within each word. To obtain syntactical embeddings for the entire sentence, we aggregate the vectors of each word in the sentence, thereby encapsulating the overall syntactical characteristics of the sentence. We make use of the available NRCLex library to measure the emotional affect of a body of jokes. We get the scores for emotions such as *fear, anger, anticipation, trust, surprise, positive, negative, sadness, disgust*, and *joy* for each joke.

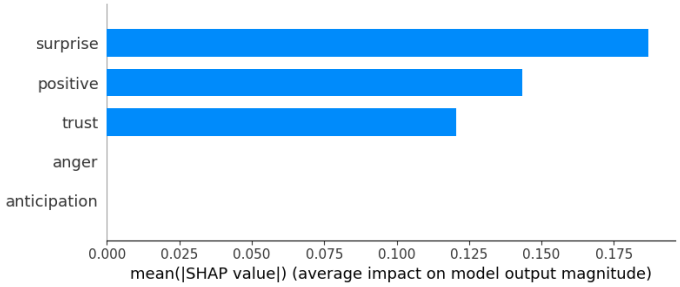

Fig. 1. SHAP on NRCLex

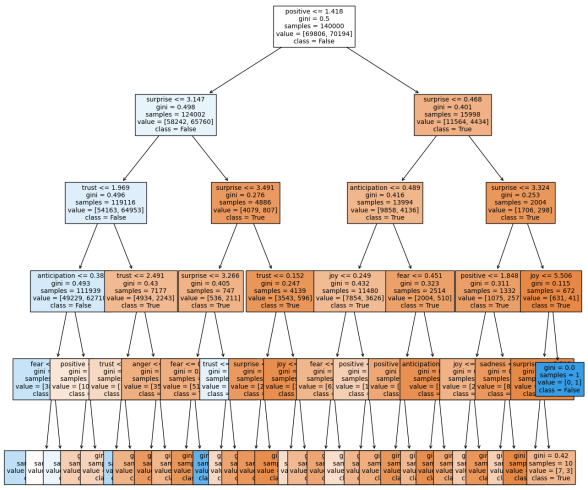

Fig. 2. DT on NRCLex

2) ***Statistics of Structural Elements****:* As described by Liu et al.[6] we follow the methods to acquire syntactic structures for humor recognition. Among these methods, we intend to focus on utilizing statistics related to structural elements to generate features for subsequent classification. These features encompass complexity metrics, phrase length ratio, average phrase length, and more, as outlined by[6] in 2018. Statistics of structural elements have been effective in evaluating the linguistic quality of text (Nenkova et al., 2010). We implement the following syntactic features:

- **Complexity Metrics**: Complexity metrics can be calculated by measuring the differences from the perspective of sentence complexity as humorous and non-humorous texts may differ in the way they express intentions. Therefore, the number of noun phrases (NP count), verb phrases(VP count), prepositional phrases (PP count), and subordinating conjunctions (SBAR count) are counted respectively as features.
- **Phrase Length Ratio**: The length ratio (LR) for PP, NP, and VP is individually calculated. This ratio represents the number of words in each phrase type divided by the total sentence length.
- **Avergae Phrase Length**: The average phrase length is determined by dividing the total number of words within each phrase by the respective number of phrase types. It's essential to note two distinct calculation methods: one (AP L1) accounts for nested phrases. For instance, in a VP phrase (VP1...(...VP2...)), the VP's length equals the sum of the lengths of VP1 and VP2. The other approach (AP L2) considers only the maximum phrase length,

where the phrase length is determined by the length of VP1.

- **Ratio of PP or NP within a VP (RP NV)**: If a VP encompasses NP or PP, this value corresponds to the average length of NP or PP divided by the length of VP.

We make use of the NLTK package to obtain Statistics of Structural Elements and these features can be called SSE features (SSE).

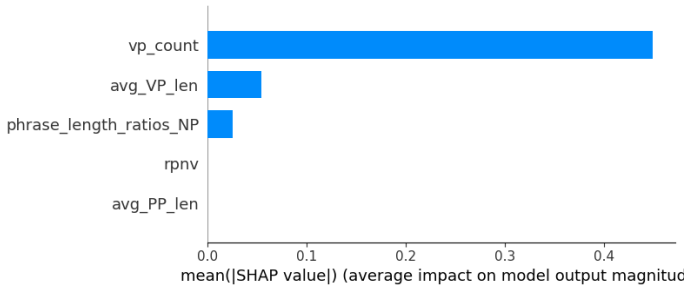

Fig. 3. SHAP on Syntactic Features

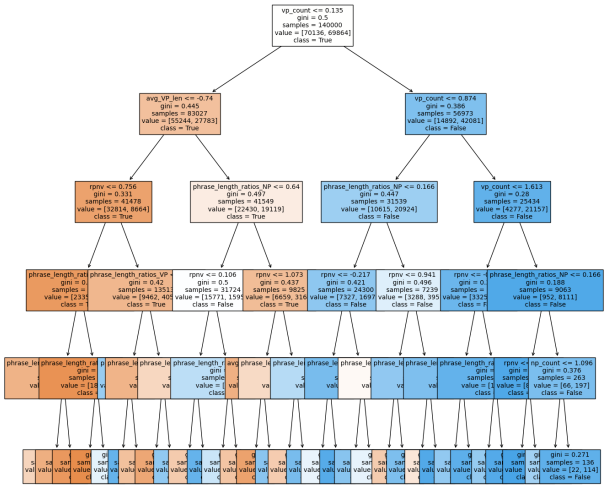

Fig. 4. DT on Syntactic Features

B. Semantic information

Among the 12 humor theories, also known as humor structures, [7] we explore the latent semantic structures behind humor in three aspects: (a) Incongruity, (b) Ambiguity (c) Phonetic Style.

- **Incongruity Structure**
  Laughter often arises from the union of seemingly contradictory or incongruous elements, forming a unique relationship or assemblage within the mind (Lefcourt, 2001). The crux of humor lies in the incongruity, the separation of one idea from another (Paulos, 2008). Humor often thrives on specific types of incongruity, like opposition or contradiction. We extract two types of features to assess the meaning distance between pairs of content words in a sentence (Mikolov et al., 2013) by leveraging Word2Vec to gauge the semantic connections within a sentence. We describe incongruity through the following two features:

  Disconnection: Representing the maximum meaning (semantic) distance among word pairs in a sentence.

  Repetition: Representing the minimum meaning (semantic) distance among word pairs in a sentence.

- **Ambiguity Theory**
  Humor and ambiguity often come together when the listener tries to interpret the meaning. Ambiguity arises when the words in a sentence can be grouped in multiple ways, resulting in various underlying interpretations. For example:
  I saw the man on the hill with the telescope.
  Different possible meanings of words lead to diverse understandings for readers. To capture this ambiguity within a sentence, we employ WordNet and assess it as follows:

  Sense Combination: This computation involves identifying Nouns, Verbs, Adjectives, and Adverbs through a POS tagger. Subsequently, we consider the potential meanings of these words *(w1, w2, ... ,wk)* via WordNet, calculating the sense combinations as

$$\log\left(\prod_{\text{sense}\in\text{senses}} \pi(\text{sense})\right) \tag{1}$$

  Sense Farmost: the largest Path Similarity of any word senses in a sentence.

  Sense Closest: the smallest Path Similarity of any word senses in a sentence.

- **Phonetic Style**
  Some studies (Mihalcea and Strapparava, 2005) suggest that the phonetic characteristics of humorous sentences hold significant importance alongside their content. Many one-liners exhibit linguistic elements like alliteration, word repetition, and rhyme, creating a comedic effect regardless of whether the joke is actually funny. An alliteration chain involves consecutive words starting with the same sound, while a rhyme chain comprises words ending with the same syllable. To extract these phonetic features, we utilize the CMU Pronouncing Dictionary and create four features:

  Alliteration: quantifies the count and maximum length of alliteration chains within a sentence.

  Rhyme: measures the count and maximum length of rhyme chains.

These can be called Human Theory Driven Features (HTF).

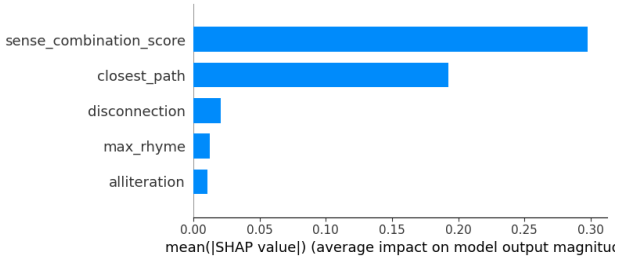

Fig. 5. SHAP on semantic features

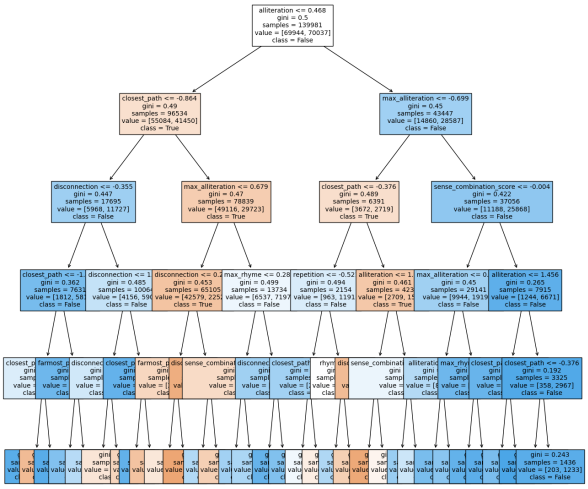

Fig. 6. DT on semantic features

C. *Contextual information*

By looking at the general structure of a joke to understand the underlying linguistic features that make a text laughable, Many suggested that humor arises from the sudden transformation of an expectation into nothing. In this way, the punchline, as the last part of a joke, destroys the perceiver's previous expectations and brings humor to its incongruity. The punchline is related to previous sentences but is included in opposition to previous lines in order to transform the reader's expectation of the context.

The proposed method for humor detection focuses on the structure of humor in text. It observes that individual sentences in a joke may seem normal and non-humorous when read separately but become humorous when considered together in context. To capture this, the method employs the model as described by Colbert[8] a neural network architecture with separate paths for sentence-level and whole-text features. The process involves tokenizing and encoding sentences using BERT sentence embeddings, followed by parallel hidden layers to extract mid-level features for each sentence.

Each sentence segment is given max_sentence_length as 20 tokens and a maximum of 5 segments are considered for an individual joke. For the whole sentence, 100 tokens are considered. The model aims to detect relationships between sentences, especially the punchline's connection to the rest, and examines word-level connections in the entire text.

These BERT sentence embeddings are generated by inputting these tokens into the BERT model, resulting in vectors of size 768. The model involves eight neural network layers, with each sentence processed through a separate parallel line of three hidden layers. These layers are concatenated in the fourth layer and continue sequentially to predict a single target value.

Parallel hidden layers in a neural network process the BERT embeddings for each sentence, extracting mid-level features related to context and sentence type, resulting in a 20-dimensional vector for each sentence.

D. *Combination of Syntactic, Semantic with ColBERT*

In our modifications we firstly combined all the previously described syntactical, semantic, and contextual information obtained from NRC, HTF into one feature list of 33 features for 200k inputs. These features were given to parallel hidden layers and then concatenated with the Bert embeddings passed through the Dense neural network layers. The model is finalized with three sequential layers of a neural network that combine the outputs from all previously hidden layers. The output was then passed through a sigmoid activation function. The model was trained with pre-trained Colbert parameters using our feature modifications. Adam optimizer and Binary Cross entropy loss were used for training and the modified model was trained for 10 epochs with a batch size of 64. These final layers aim to determine sentence congruity and detect changes in the reader's viewpoint after reading the punchline.

V. Alternatives and Justification

Our proposed method aims to assess whether a sentence is humorous or not by considering three types of information within it. While there exist traditional methods such as SVM and XGBoost,[15][16] and simpler sequence-to-sequence models like LSTM-based ones[12], the complexity of humor detection may challenge their ability to understand the full context. In contrast, more advanced models like BERT can handle this complexity better. There can also be graph-based approaches to find the relationship between sentences and words before training.

BERT [11] distinguishes different meanings for particular words based on the context by using contextualized embeddings. For example, even though the word "stick" could be both a noun as well as a verb, normal word embeddings assign

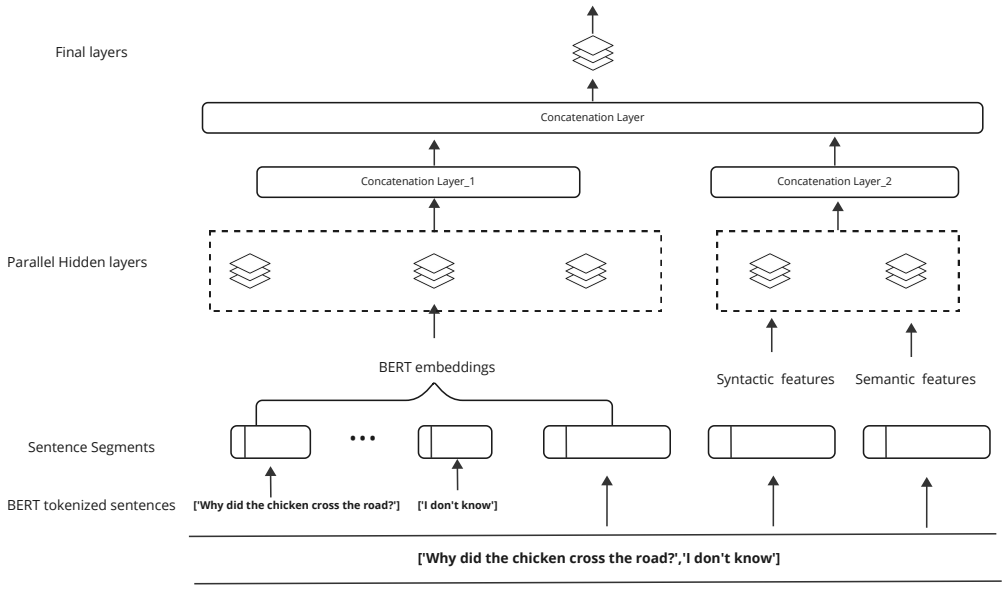

Fig. 7. Colbert Architecture with Syntactic and Semantic features

the same vector to both meanings. BERT is trained in a self-supervised way by predicting missing words in sentences, and predicting if two randomly chosen sentences are subsequent or not. Previously, there was an attempt to use BERT [9] [11] for humor detection, and it worked well on the training data. However, it didn't consider syntactical information. The only result provided was its performance on a Spanish Tweet Dataset, where it performed significantly worse (about a 15% drop).

A. Challenges Encountered

Training the COLBERT model was both interesting and challenging. The biggest hurdle was needing a lot of computer power, like high-performance GPUs. The training also took a long time because the model is complex. Making changes to COLBERT for specific tasks required tweaking its structure. These challenges taught us a lot about large language models and their model layer specifications. It also showed that managing costs is crucial. Overall, the project highlighted how vital it is for models to be both powerful and flexible.

While creating features for ambiguity, we encountered several challenges. The ambiguity structure includes the sense combination score, farmost, and closest path similarity. During the process of determining path similarity using WordNet, we grappled with utilizing the Synonym set for a given word. WordNet contains synsets for words in various forms like nouns, verbs, etc. Our objective was to exclusively identify noun meanings for a given noun, which posed a significant challenge. Obtaining information about path similarity turned out to be a time-consuming process. Programming the formulas mentioned in previous literature also presented a challenging task. Surprisingly, we anticipated that semantic features would yield higher accuracy compared to syntactic features, given our focus on understanding the meanings of words and sentences; however, the opposite turned out to be true.

VI. Evaluation

To evaluate the effectiveness of our approach for humor detection we employed a range of evaluation metrics. Accuracy, F1 score, and ROC-AUC are fundamental metrics to gauge the model's overall performance, its balance between precision and recall, and its ability to distinguish between humorous and non-humorous instances. During training we employed different methodologies, for all the models not involving BERT in any way, the data was split into train and val, and the hyperparameters were set using the val data. For BERT the whole dataset was used to finetune. To evaluate its performance on unseen data, we scraped data from a website[17] and also used a sample of jokes scraped from Reddit by a third-party[18].

A. Results

We primarily assess our hand-crafted features based on accuracy and the F1 measure. The accuracy of the combined features, encompassing all three types, ranked the highest, followed by the syntactic features. These outcomes are illustrated in the table below. Evaluating BERT and Colbert models based on accuracy, we observed a consistent trend where the performance on unseen data was lower compared to the training data. These findings are also presented in the table below.

Models	NRCLex	Syntactic	Semantic	Combined
Decision Tree	0.61	0.72	0.67	**0.74**
Gradient Boost	0.61	0.71	0.65	**0.72**

TABLE I
TRAIN DATA ACCURACY RESULTS ON HAND-CRAFTED FEATURES OF 3 TYPES AND COMBINED

Model	Without Features		With Features	
Models	Accuracy	F1	Accuracy	F1
BERT	0.50	0.37	0.52	0.39
ColBERT	0.50	0.34	**0.62**	0.60

TABLE II
RESULTS ON TEST (REDDIT) DATASET WITH AND WITHOUT FEATURES.

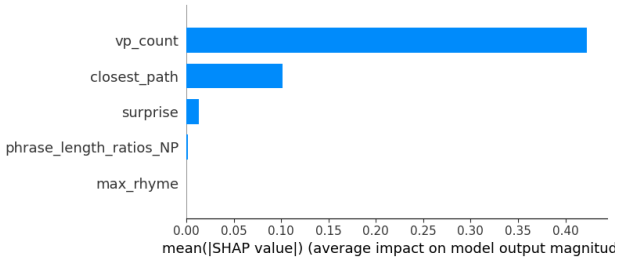

Fig. 8. SHAP on all 33 features

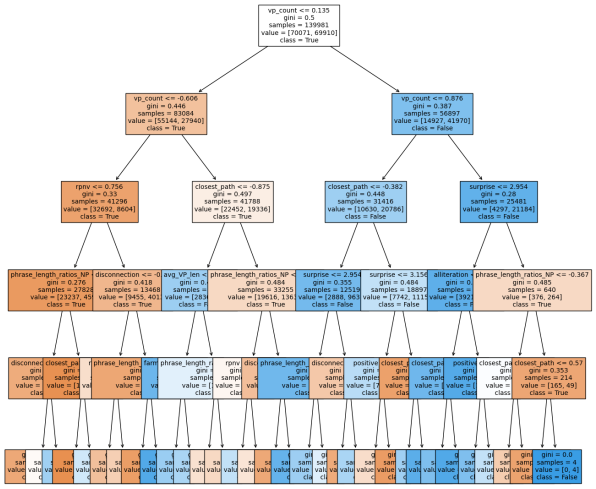

Fig. 9. DT on all 33 features

VII. MAIN FINDINGS

We've discovered intriguing nuances in how features representing emotions and semantics help the models in humor detection. Looking at the SHAP and Decision trees it was clear that it is easier for machines to detect humor based on words that exhibit emotions such as positive, anticipation, surprise, and trust. It is quite intriguing to see anticipation and surprise as they are two fundamental features that affect whether something is funny or not. Upon looking at the statistics of the syntactical elements, it seems vp_count is one feature that helps the models to understand if a sentence would be funny or not, and even after combining all the 33 features and feeding it into the model, it seems that vp_count is still what provides more understanding the model. From the features obtained from NRCLex, only the *surprise* feature seems to be present in the top 5 features, which is surprising as we expected anticipation to be present as well. But what wasn't surprising was the domination of semantic features over syntactic, which is quite understandable as you cannot take something at its face value, just because a sentence contains syntax that might contain emotions projecting humor, the whole sentence need not be humorous, compared to a sentence having a funny meaning is highly probable to be funny. But on standalone terms, embeddings that capture the contextual information, and are produced by models that are pre-trained on a huge corpus dominate every other feature. Does it mean this experimentation was a waste of time? No. Combining the above-mentioned features with the embeddings obtained from models such as BERT, resulted in higher quality which performed better on the downstream classification task, this can be seen even in the results. Overall there is comparable significance of adding syntactical and semantical elements to the contextual meanings.

VIII. CONCLUSION

In this study, we tested how well a model performs when we combine hand-crafted features with contextual embeddings. By examining jokes, we created features based on sentence structure and meaning, discovering that these features can enhance the model's ability to recognize humor. Our analysis revealed that humorous texts often: 1) use straightforward words in complex sentence structures, 2) include vivid elements like adverbs, and rhymes 3) relate closely in meaning within sentences, and 4) combine different word senses. These findings highlight the specific style elements found in humor.

REFERENCES

[1] Jurafsky, D. and Martin, J.H. (2019) Speech and Language Processing, 3rd Edition (Draft), 75-93. https://web.stanford.edu/jurafsky/slp3/ed3book_dec302020.pdf

[2] Ziser, Y., Kravi, E., & Carmel, D. (2020). Humor detection in product question answering systems. In SIGIR 2020.Retrieved from "https://www.amazon.science/publications/humor-detection-in-product-question-answering-systems"

[3] Chloé Kiddon and Yuriy Brun. 2011. That's What She Said: Double Entendre Identification. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pages 89–94, Portland, Oregon, USA. Association for Computational Linguistics.

[4] Renxian Zhang and Naishi Liu. 2014. Recognizing humor on twitter. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, pages 889–898. ACM.

[5] Amruta Purandare and Diane Litman. 2006. Humor: Prosody Analysis and Automatic Recognition for F*R*I*E*N*D*S*. In Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing,

pages 208–215, Sydney, Australia. Association for Computational Linguistics.

[6] Lizhen Liu, Donghai Zhang, and Wei Song. 2018. Exploiting Syntactic Structures for Humor Recognition. In Proceedings of the 27th International Conference on Computational Linguistics, pages 1875–1883, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

[7] Diyi Yang, Alon Lavie, Chris Dyer, and Eduard Hovy. 2015. Humor Recognition and Humor Anchor Extraction. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 2367–2376, Lisbon, Portugal. Association for Computational Linguistics.

[8] Annamoradnejad, Issa, and Gohar Zoghi. "Colbert: Using bert sentence embedding for humor detection." arXiv preprint arXiv:2004.12765 (2020)

[9] Weller, Orion, and Kevin Seppi. "Humor detection: A transformer gets the last laugh." arXiv preprint arXiv:1909.00252 (2019)

[10] Park, K., Hu, A., & Muenster, N. (2018). Laughbot: Detecting Humor in Spoken Language with Language and Audio Cues. Advances in Intelligent Systems and Computing.

[11] Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018)

[12] K. Patel, M. Mathkar, S. Maniar, A. Mehta and P. S. Natu, "To laugh or not to laugh – LSTM based humor detection approach," 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2021, pp. 1-7, doi: 10.1109/ICCCNT51525.2021.9580124.

[13] Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon, Saif Mohammad and Peter Turney, In Proceedings of the NAACL-HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, June 2010, LA, California

[14] Ani Nenkova, Jieun Chae, Annie Louis, and Emily Pitler. 2010. Structural features for predicting the linguistic quality of text - applications to machine translation, automatic summarization and human-authored text. In Empirical methods in natural language generation, pages 222–241.

[15] Tianqi Chen, Carlos Guestrin, XGBoost: A Scalable Tree Boosting System

[16] Issa Annamoradnejad, Gohar Zoghi, ColBERT: Using BERT Sentence Embedding in Parallel Neural Networks for Computational Humor

[17] Best Life Online - One Liners - https://bestlifeonline.com/funny-one-liners/

[18] orionw, "RedditHumorDetection," GitHub repository, https://github.com/orionw/RedditHumorDetection.