ASTES

Advances in Science, Technology and Engineering Systems Journal Vol. 7, No. 6, 222-229 (2022)
www.astesj.com
Special Issue on Innovation in Computing, Engineering Science & Technology

ASTES Journal
ISSN: 2205-15441

Humor Analysis via BERT-CNN Correlation based Neural Network

Rida Miraj*,[1], Masaki Aono[1], Muhammad Irfan Miraj[2]

[1] *Computer Science and Engineering, Toyohashi University of Technology, Toyohashi, Japan*

[2] *Computer Science and Engineering, Superior University, Lahore, Pakistan*

ARTICLE INFO

Article history:
Received: 28 August, 2022
Accepted: 03 December, 2022
Online: 20 December, 2022

Keywords:
Embedding
BERT
RMSE
Context learning

ABSTRACT

This paper is an extension of work initially presented in the conference name [1]. The origin of jokes has gotten much attention from many researchers in recent years. To detect humor in texts is an intriguing challenge. This paper focuses on the detection of humor in brief sentences. In this paper, we discuss the detailed mechanism that can identify the fun part inside text data. To achieve our target, we take advantage of the properties of the Bidirectional Encoder Representations from the Transformers (BERT) model to learn the real context of a sentence as faithfully as possible in a context-sensitive way. In addition to using BERT, we augment our architecture with traditional embedding methods including fastText and Word2vec. Our proposed architecture having both BERT and Embedding will be elaborated on in detail. For quantitative analysis of the effectiveness of our proposed architecture, we employ Root Mean Squared (RMS) as an evaluation measure. Our architecture has reduced the Root Mean squared error by 49%.

1 Introduction

Being a crucial component of human communication, humor or jokes puts individuals at ease and contributes to a more pleasant atmosphere [2]. Humor is a way to make people happy and make them laugh. The evolution and perception of humor have received a lot of attention recently. According to several psychologists, humor enhances mental and physical health [3]. Finding humor in a text might be seen as a precursor to tackling more challenging issues. It would be wonderful to develop jokes or other forms of humor based on an understanding of which characteristics improve texts the most. Humor in texts might be difficult for academics to discern due to regional and cultural variances. Utilizing the output of a humor detector to automatically decide whether or not a text span may be taken seriously is an intriguing use case of humor in contemporary technology [2].

Finding comedy in a book might be seen as a necessary first step before tackling more challenging problems. Automatic humor identification in texts may help human-centered artificial intelligence systems like chatbots and virtual assistants. Although comedy has been studied from psychological, cognitive, and even linguistic perspectives [4], computational linguistics has not yet looked into it from this angle. What is it about amusing sentences that makes us laugh? It largely depends on a person's cultural background and environment. Standards cannot adequately define humor because it differs from country to country, region to region, and person to person. In other words, humor is universal and is interpreted differently depending on the culture [5]. Many types of humor, including irony, wordplay, metaphor, and sarcasm, call for in-depth outside knowledge. It is more difficult to detect humor because of these factors. Recent advancements in deep learning have made humor identification more practical by enabling end-to-end training with massive amounts of data without the need for manual feature extraction. There has been a lot of discussion about how to identify humor in texts. [6]. Recent developments in natural language processing and neural network architecture have benefited ext. Summarization, Natural Language Generation [7], and Text Classification [8]. These developments have allowed scientists to study common linguistic behavior.

Unsupervised representation learning has proven to be quite successful in the field of natural language processing [9]–[11]. Before fine-tuning the models or representations on subsequent tasks, these methods frequently pre-train neural networks on big unlabeled text corpora. This same high-level notion has been used to study several unsupervised pretraining goals. One such domain, humor, has drawn attention in social media, generation, and classification.

As was already said, the research community faces a special obstacle when attempting to discern comedy from a text. A task to find humor in English news headlines using micro-edits has been developed in [12] to address the issue of humor identification from

*Corresponding Author: Rida Miraj, ridamiraj974@gmail.com

a short text. They unveiled a cutting-edge dataset for computational humor research. The following choices were used to define micro changes: entity→noun, noun→noun, and verb→verb. Each amended headline was evaluated by five judges, who assigned a score between 0 (not hilarious) to 3 for each one, according to [12] (very funny). The average of each headline's five funniness grades determines how funny it is. The goal is to understand how short changes to humor can be perceived by machines. These changed headlines are known as humicroedit. The foundation for the construction of humorous writing can be laid by accurately assessing the humor in micro-edits, according to [12]. Fig. 1 demonstrates how a single word is changed to another term to make the news headline on the Reddit website amusing. In our study, we used this dataset. On this dataset, various methods were used to estimate the level of humor in those modified headlines.

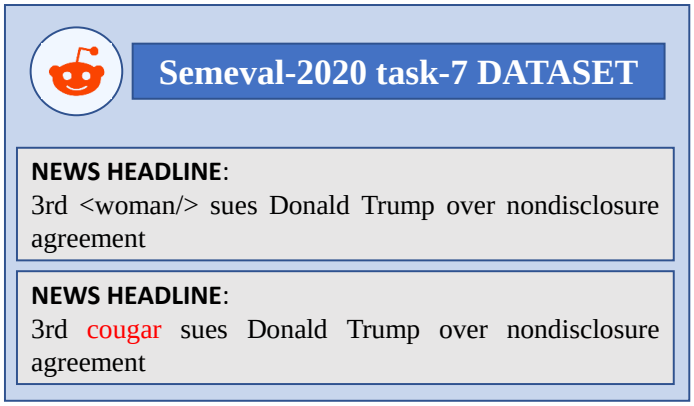

Figure 1: Example of Humicroedit from dataset.

The rest of the paper is organized as follows: **Section 2** presents a summary of previous studies. **Section 3** introduces our suggested methodology for detecting humour. Experiments and evaluations are covered in **Section 4**. **Section 5** offers some final observations as well as future research directions.

2 Related Research

In recent years, humor detection task has gained a lot of attention. A widespread and elusive occurrence of humor exists throughout the world. There is a lot of research on humor detection based on binary categorization. The degree of humor was estimated using a variety of machine and deep learning techniques. In [13], alliteration, antonyms, and adult slang are a few examples of the stylistic and lexical clues that computational approaches to comedy recognition rely on. To identify funny spoken words in a comedy television show 'FRIENDS'. Purandare and Litman [14] did a humor-prosody analysis using a supervised learning classifier. They analyzed humor for both gender 'male' and 'female' separately. In [15], a heuristic focus is provided for a site where wordplay may or may not occur by the algorithm, which learns statistical patterns of text in N-grams. An utterance that sounds similar to a word is created using a wordplay generator, and the wordplay recognizer checks the utterance's validity using N-grams. Yang proposed a Maximal Decrement method to automatically extract anchors that enable humor in a sentence. The feature extraction, after identifying various semantic structures that underline humor, is used in a computational method to detect humor [16]. To detect humor in a conversation, it is necessary to check for the other hidden meanings in a sentence. In [17], instead of directly dealing with humor recognition in a sentence, Chloe and Brun introduced sub-problems "that's what she said" problem having two distinctive features: (1) the use of code words for sexually explicit nouns; and (2) a structure that is typical of the erotic domain. Recurrent neural networks (RNNs) and convolutional neural networks (CNNs) are used by Luke de Oliveira and Alfredo to detect humor in Yelp reviews [18]. In [19], Bidirectional-LSTM (BiLSTM) model was used in the Semeval-2020 task-7 for pun detection. Orion Weller [20] employed a Transformer architecture for its advantages in learning from sentence context in detecting humorous and non-humorous jokes. The detection of humor is not limited only to English language statements, or English shows; however, a lot of work is done in other languages. Researchers are publishing more high-quality datasets in different languages which provides significant help in this area. [21, 22] examined several languages to detect humor. Some researchers focus on employing state-of-the-art studies like BERT [11] and GPT [23] to make better predictions, whereas some researcher attempt to improve lighter networks like LSTM [24] and CNN [25] or even conventional machine learning algorithms to compete with deep neural network models. Vladislav Blinov [26] utilized classification techniques, including the linear Support Vector Machine (SVM) to recognize comedy in texts.

Some studies focused on using pre-trained models for a range of text classification problems due to the popularity of transfer learning. In Natural language processing (NLP), specific models like ELMO [10] and BERT [11] are focused on transfer learning, which focuses on retaining knowledge gained through training on one issue and applying it to another, similar issue, typically after fine-tuning on a small amount of data. BERT uses a multi-layer bidirectional transformer encoder, which can learn complex bi-directional representations because it consists of several encoders stacked on top of each other. Similar to prior transfer learning systems, it is pre-trained on unlabeled data before being adjusted for a variety of tasks. This method was also applied to humorous jobs in [27], where several BERT layers were employed. There are a lot of competitions held where organizers made a challenge to detect humor from different types of the corpus. The impact of figurative language, including metaphors and irony, on sentiment analysis, was one of the difficulties described in Task 11 in Semeval-2015 [28]. In Semeval-2017 Task 6 [29], participants were given a selection of amusing tweets that had been sent to a comedy show, and they were asked to predict how well the producers and viewers of the show would have ranked the tweets. The HAHA task was organized by Grupo PLN-InCo at IberEVAL 2018 [30] and IberLEF 2019 [31] respectively. Both of them included the tasks of humor detection and funniness score prediction as their two subtasks. A combination of humor detection and offense detection is used in Task 7 [32]. The subtasks are the same as in HAHA 2018 and 2019, and two new tasks are added: Offense Score Prediction and Classification of Controversial Jokes.

The majority of similar algorithms now use neural network models to identify humor in funny tweets or funny conversation.

However, [33] evaluated humor in news headlines, which posed a slightly different kind of issue. For this, we've developed a reliable architecture. We present a system that uses multi-kernel convolution to combine BERT with numerous word embeddings in a unified architecture after fine-tuning.

3 Proposed Methodology

This section concentrates on our proposed architecture that is used to find humor in text data. The logical approach is used to fulfill our target, which can detect the level of funniness in news headlines. A detailed overview of our proposed architecture is depicted in Fig. 2. The initial step in our technique is to utilize the multi-kernel convolution filters which have three different kernel sizes, such as [2,3,4]. The plus point of using multi-kernel convolution filters is their ability to extract the high feature sequence from embedding. After retrieving the feature sequence from sentence-appended embeddings, the results are concatenated. On the other hand, fine-tuning of the BERT model is done. After the fine-tuning, all high feature information is registered in the initial token of a sentence and we save that feature vector as the output from the BERT model. This feature vector is assembled with the feature vectors of the multi-kernel convolution layer. The collective information is entered into a fully connected humor prediction system, which provides the final predictions. A detailed architecture of our system is demonstrated in Fig. 2. In the following sections, each component is described thoroughly.

3.1 BERT

BERT [34] which is known as Bidirectional Encoder Representations from Transformers, had great success in various language understanding standards. It is a new way of language representation that can capture basic knowledge of structure. As referred to in the name of BERT, it has twelve transformers with 85M weights. The architecture of BERT is also based on an embedded layer with 24M weights. By simultaneously focusing on the left and right contexts, BERT uses Transformer networks [35] to pre-train bidirectional representations in all levels. The transformer network in BERT uses an encoder with a self-attention layer. The current node can rely on both the semantics of the context and the current word because of self-attention. With more detailed analysis, it is found that BERT is also putting its valuable attentional resources looking at uninformative tokens e.g., punctuation marks, [SEP] because these tokens behave like a no-op when an attention head is focused on it. This enables an attention head to "switch off" if the linguistic structures are tuned for, but are not present in the input. Keeping in mind all the properties and advantages of BERT, we fine-tuned it on our task in a manner that can utilize the self-attention system of BERT to the full extent. Here, the word fine-tuning is referred to a technique that puts transfer learning into practice. It adapts or transforms a model that has previously been trained to perform one job to perform a second, related job. A plethora of knowledge about our language is available in BERT and BERT with fine-tuning helps us to construct a model in less time with fewer amounts of data that gives outcomes with better results.

3.2 Embeddings

Word embedding is the general term for a set of language modeling and feature learning methods in natural language processing (NLP) in which words and phrases are positioned on a real vector space. The mathematical embedding is performed from a space with as many dimensions as there are words to a continuous vector space with much lower dimensions. It is one of the most used representations of words in vocabulary because it can capture the words context inside a document. A vector representation of text data might be advantageous for learning algorithms in a variety of natural language processing (NLP) applications [36, 37]. Using word and phrase embeddings as input representations have been shown to improve performance on NLP tasks such as parsing and sentiment analysis. In linguistics, word embedding was discussed in the field of distributional semantics. It aims to quantify and classify semantic similarities between linguistic items based on the distributional properties of large samples of linguistic data. Distributional vectors or word embeddings follow the rules of distributional theory. The distributional theory states that words with comparable meanings tend to recur in similar settings, and. As a result, these vectors attempt to represent a word's neighbors' qualities. Distributional vectors' key benefit is that they can measure word similarity. In humor detection tasks, typical embedding approaches have been used in the last few years. We also utilize the benefits of word embedding in our proposed architecture, as seen in Fig.2. We use three embedding representations for news headlines which in the end are concatenated to make a unique word vector matrix. The dimensionality of a matrix is represented as $L \times D$, where L and D depict the length and dimension of the word vector, respectively. In this manner, we obtain a vector representation of words by using a pre-trained word embedding model.

3.3 Multi-Kernel Convolution

In [38], a technique is presented which is used in our multi-kernel convolution layer for higher-level features extraction. The embedding matrix generated from the embedding layer is used as input for this system. On the input, convolution is done by using filters. Three different kernel sizes are used in our system to apply convolution. The kernel sizes are 2,3, and 4. A response is generated from each filter. After performing a convolution, a feature map is formed by each filter. The output from this layer is then used to establish a univariate feature vector that uses the max-pooling function. In the last step, features from all kernels are combined to construct a new vector in which high-level features are saved.

3.4 Humor Prediction and Model Training

This is the final step that is done to get the prediction for humor detection for Edited News headlines. The feature vectors established in the BERT and embedding-based CNN model are concatenated and then passed to a fully connected linear layer. To evaluate the whole performance of our architecture, we use a loss function known as mean square error (mse). Mean squared error (mse) in statistics/machine learning is a function that calculates the square value of the "difference between the predicted value and the correct value " for each data and outputs the sum divided by the number of data.

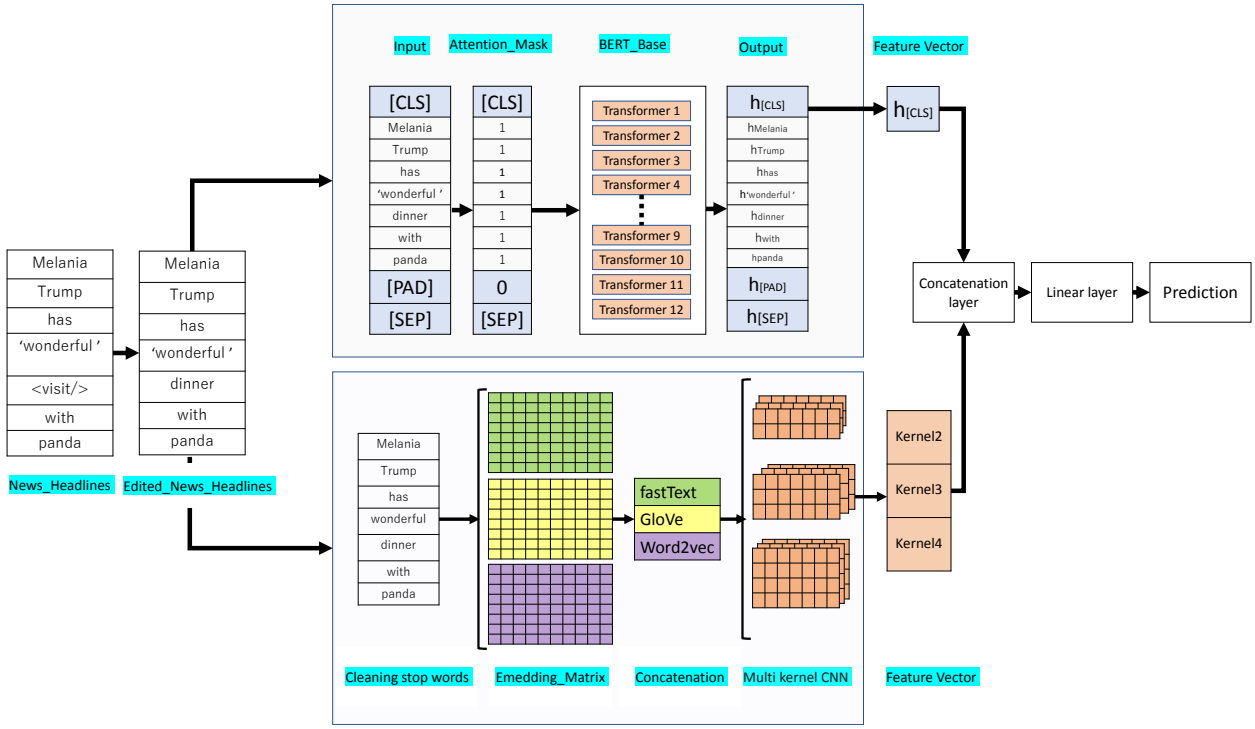

Figure 2: Proposed architecture for humor detection from edited news headlines.

The error can be "correct value - predicted value" instead of "predicted value - correct value. In the below equation, mean square error is defined as:

$$mse = \sum_{i=1}^{n} (y_i - \hat{y}_i) \quad (1)$$

$\hat{y}_i$ is the estimated value, and i is the sample index with its true value y_i. The optimizer that is used in our system is Adam optimizer [39]. Adam is now the de facto standard optimization algorithm widely used in all models. It is simply a combination of Momentum, which suppresses oscillations by moving averages, and RMSProp, which suppresses oscillations by adjusting the learning rate.

4 Experiments and Evaluation

4.1 Data

The data that is used to train and evaluate the effectiveness of our model is taken from SemEval-2020 Task 7 [33]. A detailed examination of this dataset is highlighted in this section. For the task, of humor detection, this dataset is a real challenge. This dataset is not taken from any funny website or pun-intended long scripts. It is collected from a Reddit website that contains news headlines. These news headlines are short and serious sentences, mainly about politics, war, etc. These serious headlines are alternated by one word that can make a serious sentence funny. The total distribution of the dataset is in this manner; 9652 news headlines for training data; 2419 news headlines for validation data, and 3024 news headlines for testing the model. With respect to each headline, there is a score which is the mean of labels assigned by some judge of SemEval-2020 Task 7 [33]. A score is a rational number that starts from 0 and ends at 3. Score 0 is labeled as not humorous at all and score 3 is marked as the funniest sentence. The dataset contains annotation in the form of humor assessment grades (0, 1, 2, and 3) in descending order, together with the mean of these ratings, for each sample. The pre-processing of data is done before training. There is a need to change the news headlines into Edited News headlines. For that, the replacement of one specific provided word that can put a funny impact on the whole sentence is done. In a sentence structure, the word inside these parenthesis ⟨⟩is changed with provided words. We tested several pre-processing tools to clean the data; however, last, we prefer to use raw data with minor pre-processing because our architecture response was good on it.

To clean the data, existing pre-processing software such as spaCy [40] and NLTK's [41] is used. Stopwords have a negative impact because they lack opinion-oriented information, which could hinder classifier performance. To remove stopwords NLTK's [41] standard stoplist is used. And this dataset after removal is used in one side of the architecture, which contains an embedding-based CNN model. On the other side, we are using the BERT model in our system, using stopwords elimination can affect the performance because BERT has the property to use a self-attention mechanism on these tokens. The stopwords are not removed from the dataset. The padding is required to use BERT properties. The length of a sentence can vary because of that we need to equalize the tokens of every sentence by adding padding to it.To add padding in a sentence,

a zero token with maximum length has been concatenated with each sentence. After padding, sentences are converted into vectors of a predetermined length. In our architecture, we set the padding value to 40 as it is the maximum number of words in a sentence present in our dataset.

4.2 Model Configuration

This section contains information about parameter settings and models which are used in our architecture. Starting from the right part of the Fig. 2, we utilized the following embedding models for word embedding in our system, and their properties are mentioned, the Table 1.

- fastText embedding model is pre-trained on Wikipedia [42] and serves as a library for effective word representation and text classification learning. Both supervised (classifications) and unsupervised (embedding) word and phrase representations are supported by fastText.
- GloVe embedding model [43] uses an algorithm, unsupervised learning, to learn word vector representations. To do this, words are mapped into a meaningful space where the separation between words is correlated with their semantic similarity.
- Word2Vec [44] is pre-trained on part of the Google News dataset. It is a method aimed at acquiring distributed expressions of words. It uses CBOW (continuous bag-of-words) [44] and skip-gram [45].

Table 1: EMBEDDING MODEL PROPERTIES

Embedding model	Dimension	Model Properties
fastText	300-dimensional	16B tokens
GloVe	300-dimensional	2.2M vocab and 840B tokens
Word2Vec	300-dimensional	3M vocab

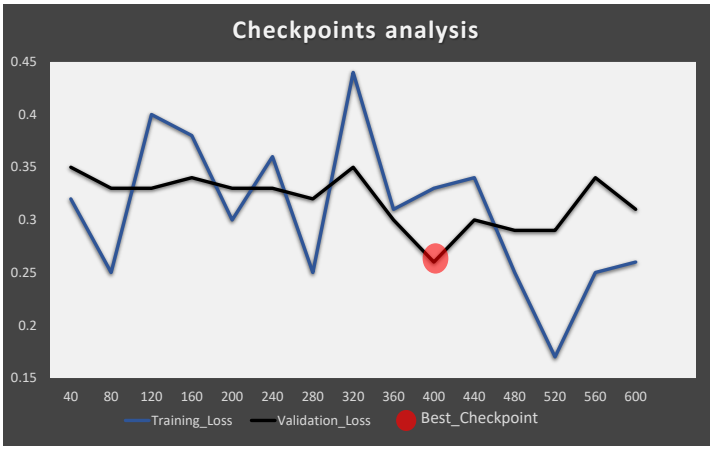

Figure 3: Parameters Information. x-axis: steps(epochs), y-axis:loss-score

After utilizing the embedding models, we employed a total of 36 filters and three kernel sizes—2, 3, and 4—for the multi-kernel convolution. On the left side of the architecture, we used the BERT model. BERT has two models, the BERT-Base model and the BERT-Large model. In our architecture, we employed a BERT-Base model keeping the size of the whole architecture in mind. BERT-Base is made of 12 transformer layers which are stacked on each other with embedding layers. The length of BERT-Base embedding with almost 30k words is 768. BERT uses a specific pattern of input data in it. To create the specific pattern, unique tokens named [CLS] and [SEP] are introduced. With that, there is a need for consistent padding length for all sentences before sending them as input into BERT. As mentioned before, we set the padding length as 40. After these constraints of BERT, an Attention Mask is implemented to make a distinguisher between actual words tokens and padded tokens. Actual words are set as 1 whereas padded ones are set as 0 so that the BERT model can focus on only the actual sentence token. Once the data is ready according to BERT requirement, it is sent in the BERT model where it passes from all 12 layers of the transformer, saving all the characteristics of each transformer layer in the first token [CLS]. After receiving the output from BERT, other tokens except [CLS] are discarded. Only the [CLS] token is used in the next steps. As BERT is a pre-trained model, we did fine-tuning of BERT on our dataset to reduce training time. For that, we used 4 number of epochs where each epoch is divided into 240 steps to analyze the performance of the model. The value of validation loss is recorded during training. After collecting all values of validation loss from training as a checkpoint, we considered the lowest value of loss validation as the best checkpoint because on this minimum value our architecture performance is better. The saved checkpoints are depicted in Fig. 3. The blue line is showing the graph of validation loss where steps are shown along the x-axis and loss values are shown on the y-axis. As it is clear in the figure that in the 2nd epoch around 400 steps, the value of loss validation is the minimum. We used this point for evaluation purposes. The "Best Checkpoint" methodology relies on the assumption that the model version with the lowest validation loss is the "finest" model, superior to the "fully-trained" model after the four epochs due to its exceptional generalizability. Moving towards the use of libraries during coding, we have used the hugging-face library [46] for BERT fine-tuning. The whole framework is established on Pytorch [47]. With that, we used GPU [48] for training the model to take the benefits of parallel tensor processing. In the end, the Adam optimizer is used during training with a batch size of 32 and an initial learning rate of 0.001. As mentioned before, for evaluation, the root mean square (rmse) is used.

$$rmse = \sqrt{(\frac{1}{n})\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (2)$$

Up to this point, we have described all kinds of parameter settings that were used to establish our system.

4.3 Results

The purpose of the creation of our architecture is to determine the level of humor in Reddit news headlines that are not supposed to be

funny content. Before establishing deep learning-based architecture, we conducted the experiment with machine learning models, the Naive Bayes algorithm, and Linear Regression and we found out that on this type of dataset, these model does not work properly. These models give almost the same prediction score for most of the sentences because these models cannot learn the context hidden between lines. More ever, after testing machine learning models, we stepped up and used BiLSTM models [19] which showed better performance than machine learning models. We employed this model with a multi-kernel convolution network in the Spanish tweet dataset [49] which gave 74 percent accuracy for humor detection in Spanish tweets.

During the creation of our architecture, we tried several combinations of embedding models with or without BERT. We have summarized a few of our approaches in which we tried to achieve our target in this Table 2. To extract the best from all pre-trained models, a lot of different hyperparameter settings are also done. The prediction score in the form of root mean square is displayed on the right side of the column. As clearly shown in the table, the best score for the humor detection challenge is using the method we introduced in this paper. Other than that, we employed Bidirectional LSTM with different embedding models to analyze the properties of BiLSTM in detecting humor in headlines.

Table 2: Comparative results with different experimental settings

Embedding	Model	*RMSE*
Baseline		.5750
Our Architecture		**.5116**
GloVe	BiLSTM	.6291
GloVe+fastText	BiLS TM	.6172
BERT Embedding	BERT (Without Finetune)	.5779
BERT-Large Embedding	BERT-Large(Finetune)	.5401
BERT-Base Embedding	BERT-Base(Finetune)	.5701

This table shows how fine-tuning the BERT model helps in giving better results rather than using the BERT without fine-tuning. We conducted experiments with the BERT-Large model, from which we found out the pros and cons of using BERT-Large. BERT-Large may perform well if we use it in our architecture in place of BERT-Base, however, due to its comparatively large size with 24 transformer layers, it becomes unsuitable for our architecture. The minimum root mean square error is recorded using our architecture.

Next, we checked our architecture performance on test data provided in Semeval-2020 task-7. As mentioned earlier, the data is collected from headlines which are then micro-edited to make them funny. We tested our architecture on Edited News headlines and simple News headlines with micro-edit. In Fig. 4, we have shown the performance of our architecture on News headlines and Edited News headlines with the ground truth score provided by Semeval-2020 organizer. The number of sentences is shown along the y-axis and a prediction score from 0 to 3 is shown along the x-axis. The gray curve in the figure shows the ground-truth (true label for each sentence) whereas purple and green curves depict our architecture performance on Edited News headlines and simple headlines, respectively. In other words, the graphs depict the number of sentences on the prediction scale [0-3].

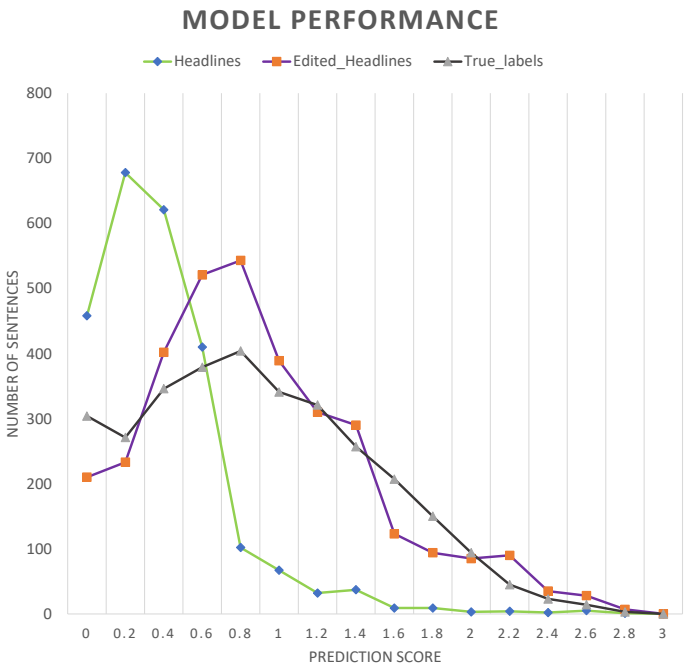

Figure 4: Model Performance on Test Data

4.4 *Further Analysis*

To show the performance of our architecture, we have shown the top results of other teams from Semeval-2020 task 7 competition. As shown in Table 3, the team Hitachi with the best results exploited the PLMs BERT, GPT-2, RoBERTa, XLNet, Transformer-XL, and XLM. Our architecture ranked in 3rd place with 0.511 RMSE with the advantage of comprehensive and less memory consumption architecture. Using our proposed architecture, our method is checking the presence and impact of each word in a sentence which results in an authentic result for humor detection.

Table 3: Semeval-2020 task-7 Top 4 teams results

Team Name	*RMSE*
Hitachi	0.49725
Amobee	0.50726
YNU-HPCC	0.51737
MLEngineer	0.51966

There are a few limitations that we observed during experiments. The first hindrance was the length of the sentence. The data is collected from headlines, and headlines tend to be small, comprehensive, and serious. The smallest length of a sentence found in our dataset is 4. For natural language processing tasks, there is a need for enough information from a sentence to extract the main sentiments from the sentence. The second limitation was a cultural difference. Due to the lack of awareness of cultural metaphors, it is hard to make a correct judgment about a sentence. Every area has its

ways of making jokes which may be hard for others to understand as a joke. One more parameter, that is affecting our system is a minor line between sarcasm and humor. The model does not seem to understand the harmful effects of negative emotions on humor or the subtle differences between sarcasm and irony.

5 Conclusion

In this study, we developed a technique for assessing the level of comedy in written sentences. In our architecture, we used a variety of deep learning techniques, such as multi-kernel convolution and BERT. Text data can be understood better when BERT and external embeddings are used with CNN models. The results prove how well our system works. The main contribution of our system is that it successfully learns contextual information, which improves humor detection performance. Even though we have competitive results, our strategy might still use some work. Our system will be more effective at classifying a sentence as amusing the more we understand what comedy is and how culture affects it. In the future, we want to focus on certain types of comedy, such as superiority, sarcasm, irony, and incongruity. This could help us comprehend how different modeling approaches might uncover diverse fundamental sources of humor.

Acknowledgment This research was partially supported by Grants-in-Aid for Scientific Research issue number 22K12040.

References

[1] R. Miraj, M. Aono, "Humor Detection Using a Bidirectional Encoder Representations from Transformers (BERT) based Neural Ensemble Model," in 2021 8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA), 1–6, 2021, doi:10.1109/ICAICTA53211.2021.9640260.

[2] S. Castro, M. Cubero, D. Garat, G. Moncecchi, "Is this a joke? detecting humor in spanish tweets," in Ibero-American Conference on Artificial Intelligence, 139–150, Springer, 2016.

[3] F. D. Ganz, J. M. Jacobs, "The effect of humor on elder mental and physical health," Geriatric Nursing, **35**(3), 205–211, 2014.

[4] V. Raskin, "Linguistic heuristics of humor: a script-based semantic approach," International journal of the sociology of language, **1987**(65), 11–26, 1987.

[5] R. A. Martin, N. A. Kuiper, L. J. Olinger, K. A. Dance, "Humor, coping with stress, self-concept, and psychological well-being," Humor, **6**, 89–89, 1993.

[6] F. Barbieri, H. Saggion, "Automatic Detection of Irony and Humour in Twitter." in ICCC, 155–162, 2014.

[7] K. Ethayarajh, "How contextual are contextualized word representations? comparing the geometry of BERT, ELMo, and GPT-2 embeddings," arXiv preprint arXiv:1909.00512, 2019.

[8] H. Bahuleyan, "Natural language generation with neural variational models," arXiv preprint arXiv:1808.09012, 2018.

[9] A. M. Dai, Q. V. Le, "Semi-supervised sequence learning," arXiv preprint arXiv:1511.01432, 2015.

[10] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, "Deep contextualized word representations," arXiv preprint arXiv:1802.05365, 2018.

[11] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv:1810.04805, 2018.

[12] N. Hossain, J. Krumm, M. Gamon, ""President Vows to Cut <Taxes> Hair": Dataset and Analysis of Creative Text Editing for Humorous Headlines," (iv), 133–142, 2019, doi:10.18653/v1/n19-1012.

[13] R. Mihalcea, C. Strapparava, "Making computers laugh: Investigations in automatic humor recognition," in Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, 531–538, 2005.

[14] A. Purandare, D. Litman, "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, 208–215, 2006.

[15] J. M. Taylor, L. J. Mazlack, "Humorous wordplay recognition," Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern., **4**(September), 3306–3311, 2004, doi:10.1109/ICSMC.2004.1400851.

[16] D. Yang, A. Lavie, C. Dyer, E. Hovy, "Humor recognition and humor anchor extraction," in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2367–2376, 2015.

[17] C. Kiddon, Y. Brun, "That's what she said: double entendre identification," in Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, 89–94, 2011.

[18] L. De Oliveira, A. L. Rodrigo, "Humor detection in yelp reviews," Retrieved on December, **15**, 2019, 2015.

[19] R. Miraj, M. Aono, "KDEhumor at SemEval-2020 Task 7: A Neural Network Model for Detecting Funniness in Dataset Humicroedit," in Proceedings of the Fourteenth Workshop on Semantic Evaluation, 852–857, International Committee for Computational Linguistics, Barcelona (online), 2020, doi:10.18653/v1/2020.semeval-1.107.

[20] O. Weller, K. Seppi, "Humor detection: A transformer gets the last laugh," arXiv preprint arXiv:1909.00252, 2019.

[21] J. A. Gonzalez, L.-F. Hurtado, F. Pla, "TWilBert: Pre-trained deep bidirectional transformers for Spanish Twitter," Neurocomputing, **426**, 58–69, 2021.

[22] B. Farzin, P. Czapla, J. Howard, "Applying a Pre-trained Language Model to Spanish Twitter Humor Prediction," arXiv preprint arXiv:1907.03187, 2019.

[23] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al., "Language models are unsupervised multitask learners," OpenAI blog, **1**(8), 9, 2019.

[24] S. Hochreiter, J. Schmidhuber, "LSTM can solve hard long time lag problems," Advances in neural information processing systems, **9**, 1996.

[25] A. Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, **25**, 2012.

[26] V. Blinov, V. Bolotova-Baranova, P. Braslavski, "Large dataset and language model fun-tuning for humor recognition," in Proceedings of the 57th annual meeting of the association for computational linguistics, 4027–4032, 2019.

[27] R. Miraj, M. Aono, "Integrating extracted information from bert and multiple embedding methods with the deep neural network for humour detection," arXiv preprint arXiv:2105.05112, 2021.

[28] A. Ghosh, G. Li, T. Veale, P. Rosso, E. Shutova, J. Barnden, A. Reyes, "Semeval-2015 task 11: Sentiment analysis of figurative language in twitter," in Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015), 470–478, 2015.

[29] "#HashtagWars: Learning a Sense of Humor ¡ SemEval-2017 Task 6," .

[30] S. Castro, L. Chiruzzo, A. Rosá, "Overview of the HAHA task: Humor analysis based on human annotation at IberEval 2018." in IberEval@ SEPLN, 187–194, 2018.

[31] L. Chiruzzo, S. Castro, M. Etcheverry, D. Garat, J. J. Prada, A. Rosá, "Overview of haha at iberlef 2019: Humor analysis based on human annotation," in IberLEF@ SEPLN, 2019.

[32] J. Meaney, S. Wilson, L. Chiruzzo, A. Lopez, W. Magdy, "Semeval 2021 task 7: Hahackathon, detecting and rating humor and offense," in Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), 105–119, 2021.

[33] N. Hossain, J. Krumm, M. Gamon, H. Kautz, M. Corporation, "SemEval-2020 Task 7: Assessing Humor in Edited News Headlines," (2019), 2020.

[34] J. Devlin, M. W. Chang, K. Lee, K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, **1**(Mlm), 4171–4186, 2019.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, "Attention is all you need," arXiv preprint arXiv:1706.03762, 2017.

[36] A. Narayanan, M. Chandramohan, R. Venkatesan, L. Chen, Y. Liu, S. Jaiswal, "graph2vec: Learning distributed representations of graphs," arXiv preprint arXiv:1707.05005, 2017.

[37] J. Tetreault, J. Burstein, C. Leacock, H. Yannakoudakis, "Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications," in Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications, 2017.

[38] J. P. A. Vieira, R. S. Moura, "An analysis of convolutional neural networks for sentence classification," in 2017 XLIII Latin American Computer Conference (CLEI), 1–5, IEEE, 2017.

[39] Y. Kowsari, S. J. M. Chabok, M. H. Moattar, "Classification of Pulmonary Images By Using Generative Adversarial Networks," in 2020 8th Iranian Joint Congress on Fuzzy and intelligent Systems (CFIS), 133–137, IEEE, 2020.

[40] N. Colic, F. Rinaldi, "Improving spaCy dependency annotation and PoS tagging web service using independent NER services," Genomics & informatics, **17**(2), 2019.

[41] E. Klein, "Natural Language Toolkit (NLTK)," 2011.

[42] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, A. Joulin, "Advances in pre-training distributed word representations," arXiv preprint arXiv:1712.09405, 2017.

[43] J. Pennington, R. Socher, C. D. Manning, "Glove: Global vectors for word representation," in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 1532–1543, 2014.

[44] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, J. Dean, "Distributed representations of words and phrases and their compositionality," arXiv preprint arXiv:1310.4546, 2013.

[45] T. Mikolov, K. Chen, G. Corrado, J. Dean, "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.

[46] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, et al., "HuggingFace's Transformers: State-of-the-art natural language processing," arXiv preprint arXiv:1910.03771, 2019.

[47] N. Ketkar, "Introduction to pytorch," in Deep learning with python, 195–208, Springer, 2017.

[48] J. Hong, M. Stone, B. Navarrete, K. Luongo, Q. Zheng, Z. Yuan, K. Xia, N. Xu, J. Bokor, L. You, et al., "3D multilevel spin transfer torque devices," Applied Physics Letters, **112**(11), 112402, 2018.

[49] R. Miraj, M. Aono, "Humor Detection in Spanish Tweets Using Neural Network." in IberLEF@ SEPLN, 837–843, 2021.