第 48 卷 第 8 期
2021 年 8 月

湖南大学学报（自然科学版）
Journal of Hunan University（Natural Sciences）

Vol. 48. No. 8.
August 2021

Open Access Article

Deep Learning of a Pre-trained Language Model's Joke Classifier Using GPT-2

Nur Arifin Akbar[1], Irma Darmayanti[2], Suliman Mohamed Fati[3], Amgad Muneer[4]

[1] Research Department, Idenitive Mashable Prototyping, Banyumas, 53124, Indonesia

[2] Department of Informatics Engineering, Universitas AMIKOM Purwokerto, 53127, Purwokerto, Indonesia

[3] College of Computer and Information Sciences, Prince Sultan University, 11586, Riyadh, Saudi Arabia

[4] Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Perak, 32610 Malaysia

Abstract: Humor generation and classification are the hardest most challenging problems in computational Natural Language Understanding. Even humans fail at being funny and recognizing humor. This study attempts to create a joke generator using a large pre-trained language model (GPT2). Further, a jokes classifier is developed by the fine-tuning bidirectional pre-trained encoder (BERT) to classify the generated jokes and attempt to understand what distinguishes joke sentence(s) from non-joke sentence(s). Qualitative analysis reveals that the classifier model has specific internal attention patterns while classifying joke sentences, which is absent when classifying normal sentences. The experimental results show the superiority of the BERT model compared to CNN and RNN+ attention baselines in terms of accuracy, precision, recall, and F1-score. The BERT model has achieved an accuracy of 0.983, precision (0.953), recall (0.978), and F1-score (0.964)

Keywords: Deep Learning, Pre-trained, Joke Classifier, Generative Pre-trained Transformer 2 (GPT-2), Bidirectional Encoder Representations from Transformers (BERT).

使用 格磷噸-2 對預訓練語言模型的笑話分類器進行深度學習

摘要: 幽默生成和分類是計算自然語言理解中最具挑戰性的問題之一。即使是人類也無法變得有趣和識別幽默。本研究嘗試使用大型預訓練語言模型 (格磷噸 2) 創建笑話生成器。此外，我們通過微調預訓練 (伯特) 來開發一個笑話分類器，以對生成的笑話進行分類，並試圖了解是什麼將笑話句子與非笑話句子區分開來。定性分析表明，分類器模型在對笑話句子進行分類時具有特定的內部注意力模式，而在對正常句子進行分類時則沒有這種模式。實驗結果表明，伯特 模型與 美國有線電視新聞網 和 循環神經網絡+ 注意力基線相比，在準確率, 準確率, 召回率和 的 1 分數方面具有優越性。伯特 模型的準確率達到了 0.983，精度 (0.953)，召回率 (0.978) 和 的 1-分數(0.964)。

關鍵詞：深度学习,预训练,笑话分类器,生成式预训练 变压器 2 (GPT-2),来自 变形金刚 的双向编码器表示 (BERT)。

Introduction

Humor is abstract, high-level use of language that is largely subjective. Even humans fail at being funny and recognizing humor. Thus, humor detection and generation continue to be challenging AI problems. This paper explores the possibilities of humor generation and classification through large pre-trained language models. Further, the authors analyzed patterns emerging when the models classify a humorous sentence and use such a model to understand how a joke differs from a normal sentence. While there have been some advances in automatic humor recognition [1, 2, 3, 4], computerized automatic humor generation has seen less progress [5, 6]. Humor generation techniques include generating humor of a specific kind, such as *I like my X like I like my Y, Z* jokes, where X, Y, and Z are variables to be filled in. Authors in [7] approached the problem through funny acronym production for given sentences. Also, the current baseline models are based on the convolutional neural network (CNN) and recurrent

Received: May 7, 2021 / Revised: June 6, 2021 / Accepted: July 11, 2021 / Published: August 30, 2021

About the authors: Nur Arifin Akbar, Research Department, Idenitive Mashable Prototyping, Banyumas, Indonesia; Irma Darmayanti, Department of Informatics Engineering, Universitas AMIKOM Purwokerto, Purwokerto, Indonesia; Suliman Mohamed Fati, College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia; Amgad Muneer, Department of Computer and Information Sciences, Universiti Teknologi PETRONAS, Perak, Malaysia

neural network (RNN) with attention and do not utilize recent advances like transfer learning [8] or even large pre-trained models like BERT [9] or XLNet [10]. In this study, to the best of the authors' knowledge, there has not been any work performed to create generic humor that does not stick to any specific humor stereotype or strict semantics. The authors overcome this limitation by relying strictly on unsupervised humor generation. In addition, there seems to be a severe lack in studying humor via computational models. Both concerns have been addressed in this paper. Mainly, the authors have verified the claim that humor has a punchline in the end that strongly connects with what is mentioned at the start of the sentence, which elicits surprise to the reader, thereby making it funny [11]. Further, the attention mechanism of the transformers has been investigated to verify this claim.

Recently, unsupervised generative pre-trained transformer 2 (GPT-2) models have promised language generation tasks greatly [12]. The authors' approach relies on fine-tuning large unsupervised language modes like GPT-2 on humor data. The authors developed a humor classifier based on BERT [9] by fine-tuning many jokes and non-jokes. Such an approach has been proven to succeed in literature because of their effective pre-training and their capacity to learn contextualized representations via transformers. The authors evaluated their classifier against existing baselines for humor (Rohan Bais) and later evaluated the GPT2 generated jokes over it compared to non-fine-tuned GPT2 that generates generic sentences. As a result, the authors' humor classifier outperforms the state-of-the-art joke classifier, and therefore, is a reliable metric for the quality of the joke generated by GPT2. The authors later analyzed the attention patterns in the final layers of the classifier that are specific to joke sentences to study the joke semantics.

Therefore, the proposed generative model is fine-tuned over the short-jokes dataset. Due to the subjective nature of jokes, the authors decided to go with jokes comprising two sentences or less as they had a better probability of being funny. Therefore, each input instance to the model is one joke followed by the "end of text" tag. In this study, the authors' contribution can be summarized as follows:

1. Unsupervised humor generation is still a challenge, but the jokes generated from newer language models like GPT2 significantly improve the quality of joke generation compared with existing methods.
2. The authors developed a state-of-the-art humor classifier based-BERT that is later used to evaluate the jokes generated by GPT2.
3. The authors analyzed the attention pattern in their joke classifier and found visible patterns for jokes compared with sentences that are not jokes with remarkable consistency. Later the authors use this to validate the joke hypothesis that every joke should have a surprise-inducing element that contradicts something mentioned at the beginning of the same sentence(s).

The rest of this paper is structured as follows. Section 2 highlighted the background and the related works of the study. Section 3 outlines this work's research methodology, while Section 4 describes the experimental findings. Lastly, section 5 concludes the paper and highlights the future work.

1. The Task

The authors implement a state-of-the-art joke classifier by fine-tuning the pre-trained BERT model proposed by [13] with custom-generated data for non-jokes. The authors then implement a joke generator by fine-tuning GPT2 that is evaluated against this classifier. Challenges for joke generation include controlling the generation via different sampling techniques and imposing penalties on bad token generation. Unlike the previous joke generation methods, the authors aim to create a generic joke that does not follow a particular joke pattern.

1.1. Joke Generation

The authors use GPT2 as a pre-trained model for joke generation. Then, the authors give a high-level overview of the fine-tuning performed over GPT2. The details are described in section 3.

The fine-tuning task on GPT2 is set up as a problem of language model fine-tuning with the short- jokes dataset. First, the authors try to predict the next word of a sentence in an auto-regressive manner. Then, from the dataset, the authors remove jokes that are more than two sentences in length to evaluate the classifier efficiently since most of the joke-no joke data the authors have followed this pattern. Next, the authors control the joke generation using sampling methods like top-k [10] and nucleus sampling (top-p) [14]. The authors also penalize repeated generation of the same tokens [15].

1.2. Joke Classification

For the joke classification task, the authors propose two baseline models. This section gives a brief description of these models and the authors' issues to improve.

1.2.1. Baseline Model

CNN has gained massive success in image classification [16], computer vision [17], and sequence prediction [18, 19]. This study utilized a CNN model, which shows success in several text categorization tasks [20]. The model represents sentences as stacked word embedding. It will retrieve a matrix of word embedding representing a sentence before performing convolutions on this matrix. The authors have used convolutions on regions of 2, 3, 4, and 5 words. The idea is to capture adjacent words in a "window" and retrieve a vector representing the word and its neighbors via convolutions. Then, the authors perform a max-pool on the outputs, concatenate the pooling into a vector, and feed into a fully connected layer that outputs the probability of the two classes as presented in Fig. 1.

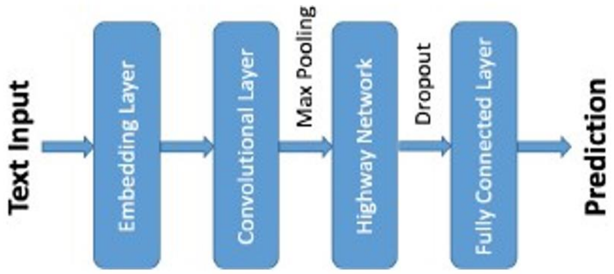


Fig. 1 CNN network architecture

Second, the authors utilized an RNN model with attention. The sentence has been fed into an embedding layer and passes each embedding as inputs to the RNN. For each timestamp, it returned to a hidden state. In order to calculate attention to find out which of the words are more important for each hidden state, the authors have fed each hidden state into a multiplicative attention mechanism. The RNN is similar to Oliviera's model [21]. Only an attention mechanism is added to improve the vanilla RNN data fitting.

2.2.2. Drawbacks of Baseline(s)

The proposed baselines model captured different features of the sentence structure individually. For example, CNN captured word window features while RNN with attention can capture time-based data retention. Nevertheless, they were not complex enough to capture both features at the same time. Additionally, they have inherent limitations in language model pre-training. It has been observed that transformer-based models outperform most other models in almost every natural language processing (NLP) task due to their advanced context representation capabilities.

Table 1 GPT2 generated jokes and non-jokes along with their classification scores

S.No.	Prompt	Type	Output	Score
1.	When did Obama become	Joke	When did Obama become the famous black guy? When he was in prison.	0.999
		Non-Joke	When did Obama become president of the USA?	0.356
2.	I love NLP	Joke	I love NLP. It is the only class I can do.	0.985
		Non-Joke	I love NLP, even though I do not know what it is.	0.568
3.	When is	Joke	When is a door not a door? When it is ajar.	0.985
		Non-Joke	When is the new moon?	0.312
4.	Why did Aditya cross the road?	Joke	Why did Aditya cross the road? To get to the other bride.	0.912
		Non-Joke	Why did Aditya cross the road? Aditya: I did not cross the road.	0.317
5.	What do you call a	Joke	What do you call a Dinosaur with Diarrhea? A shitty-sore-ass	0.998
		Non-Joke	What do you call a person with no name?	0.671
6.	Joe stood up and spoke to the	Joke	Joe stood up, spoke to the man in the wheelchair, and said I would not stand up for you.	0.015
		Non-Joke	Joe stood up and spoke to the crowd.	0.008
7.	Obama flew to Mexico	Joke	Obama flew to Mexico and asked for a Juan	0.612
		Non-Joke	Obama flew to Mexico.	0.471
8.	Isn't language learning	Joke	Is not language learning like being a prostitute? It is much work for a prostitute.	0.999
		Non-Joke	Is not language learning is a good thing. It is a good thing.	0.879
9.	When did Aditi	Joke	When did Aditi go to college? At the end of the day.	0.872
		Non-Joke	When did Aditi tell you?	0.448
10.	Do you know what happened to	Joke	Do you know what happened to the guy who got his left side cut off? He is all right now.	0.998
		Non-Joke	Do you know what happened to the other guys?	0.034

2. Proposed Approach

Generative pre-training of sentence encoders is shown promising results in NLP tasks like sentence generation and classification [25-27]. The authors' experiments use pre-trained transformer models from Hugging Face [13] to fine-tune them for joke generation and classification, respectively.

2.1. Fine-Tuning and Joke Generation Using GPT2

The authors use distilGPT2 as a pre-trained transformer model due to its smaller parameter size. GPT2 is based on the original GPT2 but has reduced parameter size due to knowledge distillation. GPT2 is a large unsupervised Transformer language model. It was trained on the WebText corpus, which contains slightly over 8 million documents with a total of 40 GB of text obtained from URLs in Reddit submissions with at least three upvotes.

GPT2 uses byte-pair sentence encodings. The sentence embedding has a dimension of 768. The model consists of six layers with twelve attention heads each. The authors limit the sequence length to 30 for the task. Any sentence that is longer than that is ignored. Suppose a sentence is smaller than 30 tokens long, the authors' pad <endoftext> at the end. The generation is controlled via top-k and top-p sampling. The authors use a top-k value of 50 and pick only the top 50 logits from the model output. Out of these 50 tokens, the authors pick the highest tokens, which sum to a total top p-value of 0.8. A repetition penalty conducted by [15] of 1/1.2 is multiplied with the logits to penalize the repeated generation of the same tokens repeatedly. Table 1 provides the results obtained. The proposed model architecture is shown in Fig. 2. A prompt is fed to the fine-tuned GPT2 model, which completes the sentence by making it funny. This completed sentence is then fed to the joke classifier, classifying it as either joke or non-joke. The authors also analyze the attention of each head of the last two layers of the classifier.

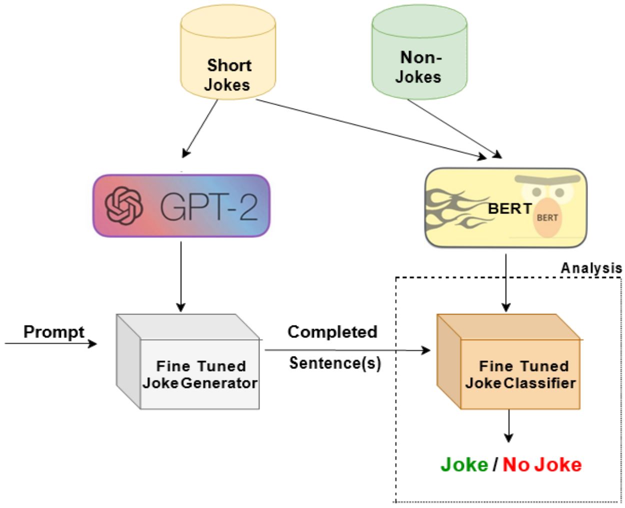


Fig. 2 The proposed model architecture

2.2. Implementing the Joke Classifier

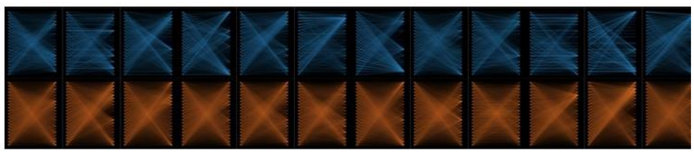
The authors have selected BERT's base uncased as a pre-trained model. BERT is a multi-layer bidirectional transformer encoder and was initially trained on a 3.3-billion-word corpus. The authors then fine-tune it with the joke-no joke training set to output probability for the two classes. The authors have limited the sequence length to 42 and have trimmed the sentences, which were longer. The authors chose to use this transformer-based model because of its success at recognizing and attending to the most important words in both sentence and paragraph structures. The authors have trained a model for three epochs.

2.3. Analysis of Attention Values of the Classifier

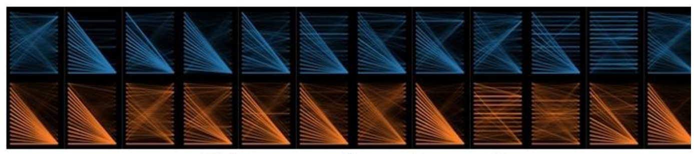
An excellent way to understand the transformer is by visualizing the attention patterns for individual attention heads in the model. The attention mapping of the sentences generated by the classifier could be visualized based on the magnitude of the attention values, as shown in Fig. 3. It would indicate which word pairs have the strongest relation while capturing the sentence semantics as a whole. The attention values of the classifier are visualized using [18]. It was found that there is a visible pattern in the shape of 'X' towards the last two layers of the classifier, which is absent for non-joke sentences. This finding indicates that the words in phrases occurring at the end of the sentence strongly connect with what was mentioned at the beginning of the sentence.

Fig. 3 Sample attention pattern for one single head. The thickness of the lines indicates how strong the attention is in between two words of the same sentence.

This result validates the claim that a joke generally consists of "setup" followed by a sudden surprise element at the end relating to the start of the sentence [11]. This pattern is observed with strong regularity for jokes and hints that for a sentence(s) to be a joke, there should be a word or a phrase at the beginning of a sentence that the end is referring to. This pattern is not observed for non-joke sentences, as presented in Fig. 4



(a) Attention pattern for joke sentence(s)



(b) Attention pattern for non-joke sentence(s)

Fig. 4 Visualizing attention patterns of the last two layers of the joke classifier. Each layer has 12 heads. It can be seen that their attention when visualized, looks like an 'X' shape for joke sentences. This observation hints that the end of the sentence has a strong connection/association with the beginning of the sentence. This pattern is not observed in normal sentences.

3. Proposed Model Evaluation

The proposed classifier is evaluated against the baseline models. The experimental results show that the proposed classifier outperforms the baseline models with an accuracy of 0.983, as shown in Table 2. Hence, GPT2 generated sentences were evaluated using this classifier.

The authors assume that the sentences generated using a fine-tuned model are a joke during the classification task, whereas the pre-trained model generated is a generic non-joke sentence.

The classifier gives an accuracy of 0.74, which is a surprisingly good score for an unsupervised genitive model. The attention analysis for a given sentence is based on the attention heads of the classifier while it tries to classify a sentence as a joke or no-joke.

3.1. Dataset Details

To evaluate the performance on humor recognition, the authors need the dataset to consist of both joke (positive) and non-joke (negative) samples. The authors used the Short Jokes (Moudgil) dataset for the positive samples, and for the negative samples, the authors chose WMT162 English news crawl as the authors' non-joke data resource.

4.1.1. Short Jokes Dataset

The short jokes dataset has been taken from an open database on a Kaggle project (Moudgil). It contains 231,657 short jokes with no restriction on joke types scraped from various joke websites and lengths ranging from 10 to 200 characters.

3.1.2. WMT162 English News Crawl

The authors choose WMT162 English news crawl as non-humorous data, similarly utilized by [19]. However, simply treating sentences from the resource as negative samples could result in deceptively high classification performance due to the domain differences between positive and negative data. Therefore, the authors are trying to pick examples whose words all appear in the positive samples and whose word count equals the average text length of the joke sentence. This was done to match the two halves (jokes and non-jokes) as closely as possible.

3.2. Evaluation Measures

The authors used four well-known evaluation matrices to measure the effectiveness of the classifier since data is balanced. This metrics is described as follows:

3.2.1. Accuracy

Accuracy denotes the ratio of the true detected cases to the overall cases, and it has been utilized to evaluate models on balance datasets [24]. Therefore, it can be calculated as follows:

$$Accuracy = \frac{(tp+tn)}{(tp+fp+tn+fn)} \quad (1)$$

where *tp* means true positive, *tn* is a true negative, *fp* denotes false positive, and *fn* is a false negative.

3.2.2. Precision, Recall and F-Measure

Precision calculates the ratio of relevant jokes in the middle of the true positive (tp) and false positive (fp) jokes belong to a particular category. Recall calculates the ratio of retrieved relevant jokes over the total number of relevant jokes/sentences. F-Measure provides a way to combine both precisions and recall into a single measure that captures both properties.

$$\text{Precision} = \frac{tp}{(tp + fp)} \quad (2)$$

$$\text{Recall} = \frac{tp}{(tp + fn)} \quad (3)$$

$$\text{F-measure} = \frac{2 \times precision \times recall}{precision + recall} \quad (4)$$

Table 2 Comparison of the authors' joke classifier with the baseline models. The authors' model outperforms both the CNN and RNN+attention models

Method	Accuracy	Precision	Recall	F1
CNN	0.906	0.902	0.946	0.924
RNN + Attention	0.921	0.918	0.935	0.926
BERT	0.983	0.953	0.978	0.964

The authors ran the CNN model with windows of up to 5 words and used 1-dimensional convolutions with max-pooling. The results are concatenated before feeding into a fully connected layer. Both CNN and attention-based RNN models were run for 50 epochs. By trying out different batch sizes, learning rates, and dropout rates, the authors found that the ideal hyper-parameters were learning rate of 0.1, batch size of 32, dropout of 0.3. As a result, CNN achieves a baseline accuracy of 0.906, whereas RNN + attention achieves 0.921.

4. Results and Analysis

The proposed best model based on BERT can achieve an accuracy of 0.983, a significant improvement over the baseline. The classification results of the joke classifier are reported in Table 2. The results of joke generation and classification are summarized in Table 1. The following are the authors; experimental findings.

1. The authors outperformed the current baseline for joke classification by a large margin, pointing out the effectiveness of pre-trained language models based on transformers.
2. Occasionally, GPT2 generates sentences with a question mark in the middle, which tricks BERT into classifying them as jokes. This problem could be eliminated by adversarial training.
3. The authors have demonstrated that the classifier can capture 'X' shape attention pattern for jokes sentences which validates the hypothesis that jokes generally follow "setup" and "punchline" structure wherein there would be strong associations between one or two words at the beginning of the end of a sentence. This association captures the shock/surprise, which results in humor.
4. The authors have examined if the length of the prompt given to the GPT2 makes any difference in the

effectiveness of a joke. Fig. 5 shows that the maximum accuracy is found for prompts of length 3 and 8, and the minimum is found for prompts of length 9.

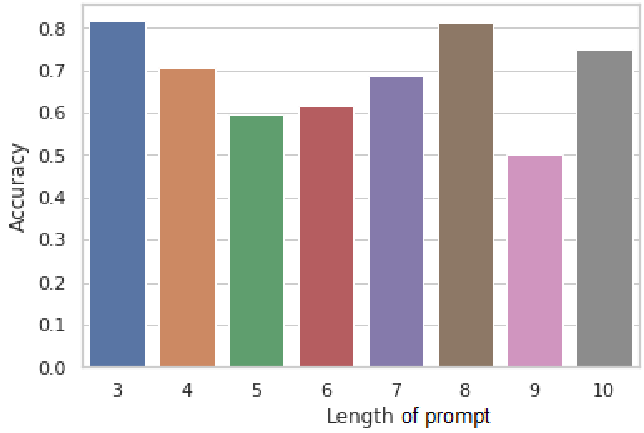


Fig. 5 Prompt length vs. accuracy of being classified as a joke

5. Conclusion

This paper developed a novel joke classifier that outperformed the previous baseline using large pre-trained models. The authors also proved that creating a high-quality joke generator without relying on hard-coding joke sentence semantics into the model by using large pre-trained language models like GPT2. The authors have also captured the hypothesis that most jokes have a setup and punchline structure compared with normal sentences by analyzing the attention patterns of the last layers of the classifier.

The authors' experiments found that GPT2 could trick BERT into classifying a sentence as a joke if it has a question mark somewhere in the second half of the sentence. Thus, the authors believe a better joke generator could be created by employing adversarial training techniques to compete against the joke classifier.

Additionally, the authors created a joke sentence from a non-joke sentence, or the authors could say they made a generic humor sentence that is not attached to certain humor stereotypes, relying on unsupervised humor generation. The model used in this study is the GPT2 model. While the classification of jokes and non-joke sentences, the authors use the BERT model. The results of the classification effectiveness of the BERT model have been previously tested in this study by comparing the results of the effectiveness test of the CNN and RNN+ models.

Due to the large amount of data that can be taken as a research sample, it is also not possible for the authors to use all existing data samples. For this reason, the authors have used the Short Jokes (Moudgil) dataset for positive joke sentences samples, and for negative samples, the authors have used news crawls English WMT162 as a non-joke data source. In the future, the authors improved the proposed model to produce joke sentences that cannot fool the BERT model in classifying a joke sentence.

Acknowledgment

The authors would like to acknowledge the support of Prince Sultan University for paying the Article Processing Charges (APC) of this publication.

References

[1] KHODAK M, SAUNSHI N, VODRAHALLI K. A large self-annotated corpus for sarcasm. *CoRR*, 2017. abs/1704.05579.

[2] DAVIDOV D, TSUR O, RAPPOPORT A. Semi-supervised recognition of sarcastic sentences in Twitter and Amazon. *Proceedings of the Fourteenth Conference on Computational Natural Language Learning, CoNLL*, 2010, 10: 107–116.

[3] BARBIERI F, SAGGION H. Modelling irony in twitter, 2010, 56–64.

[4] REYES C, BRACKETT M, RIVERS S, WHITE M, SALOVEY P. Classroom emotional climate, student engagement, and academic achievement. *Journal of Educational Psychology*, 2012, 104: 700–712.

[5] BINSTED K, PAIN H, RITCHIE G. Children's evaluation of computer-generated punning riddles. *Pragmatics Cognition*, 1997, 5.

[6] PETROVIC S, MATTHEWS D. Unsupervised joke generation from big data. *ACL*, 2013.

[7] STOCK O, STRAPPARAVA C. Getting serious about the development of computational humor, 2003, 59–64.

[8] HOWARD J, RUDER S. Fine-tuned language models for text classification. *CoRR*, 2018. abs/1801.06146.

[9] DEVLIN J, CHANG M-W, LEE K, TOUTANOVA K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint*, 2018. arXiv:1810.04805.

[10] FAN A, LEWIS M, DAUPHIN YN. Hierarchical neural story generation. *CoRR*, 2018, abs/1805.04833.

[11] ATTARDO S. Linguistic theories of humor. *Walter de Gruyter*, 2010, 1.

[12] RADFORD A, WU J, CHILD R, LUAN D, AMODEI D, SUTSKEVER I. Language models are unsupervised multitask learners, 2019.

[13] WOLF T, DEBUT L, SANH V, CHAUMOND J, DELANGUE C, MOI A, CISTAC P, RAULT T, LOUF R, FUNTOWICZ M, BREW J. Hugging Face's transformers: State-of-the-art natural language, 2019.

[14] HOLTZMAN A, BUYS J, FORBES M, CHOI Y. The curious case of neural text degeneration. *CoRR*, 2019, abs/1904.09751.

[15] KESKAR NS, MCCANN B, VARSHNEY LR, XIONG C, SOCHER R. Ctrl: A conditional transformer language model for controllable generation, 2019.

[16] MUNEER A, FATI SM. Efficient and Automated Herbs Classification Approach Based on Shape and Texture Features using Deep Learning. *IEEE Access*, 2020, 8: 196747-196764.

[17] DURAIRAJAH V, GOBEE S, MUNEER A. Automatic vision-based classification system using DNN and SVM classifiers. *2018 3rd International Conference on Control, Robotics, and Cybernetics (CRC)*, 2018, 6-14.

[18] NASEER S, ALI RF, MUNEER A, FATI SM. IAmideV-deep: Valine amination site prediction in proteins using deep learning and pseudo amino acid compositions. *Symmetry*, 2021, 13(4): 560.

[19] NASEER S, ALI RF, FATI SM, MUNEER A. iNitroY-Deep: Computational Identification of Nitrotyrosine Sites to

Supplement Carcinogenesis Studies Using Deep Learning. *IEEE Access*, 2021, 9: 73624-73640.
[20] JOHNSON R, ZHANG T. Semi-supervised convolutional neural networks for text categorization via region embedding, 2015.
[21] DE OLIVEIRA L, RODRIGO AL. Humor detection in yelp reviews, 2015.
[22] VIG J. A multiscale visualization of attention in the transformer model. *arXiv preprint,* 2019. arXiv:1906.05714.
[23] CHEN P-Y, SOO V-W. Humor recognition using deep learning. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, 2: 113–117.
[24] MUNEER A, FATI SM. A Comparative Analysis of Machine Learning Techniques for Cyberbullying Detection on Twitter. *Future Internet*, 2020, 12(11): 187.
[25] DE OLIVEIRA L, RODRIGO AL. Humor detection in yelp reviews, 2015.
[26] YANG Z, DAI Z, YANG Y, CARBONELL J, SALAKHUTDINOV R, LE QV. Xlnet: Generalized autoregressive pre-training for language understanding, 2019.
[27] RAFAY A, SULEMAN M, ALIM A. Robust Review Rating Prediction Model based on Machine and Deep Learning: Yelp Dataset. *2020 International Conference on Emerging Trends in Smart Technologies (ICETST)*, 2020, 8138-8143.

参考文:

[1] KHODAK M, SAUNSHI N, VODRAHALLI K. 一個大型自註釋的諷刺語料庫。 相關性，2017。abs/1704.05579
[2] DAVIDOV D, TSUR O, RAPPOPORT A. 推特和亞馬遜中諷刺句子的半監督識別。第十四屆計算自然語言學習會議論文集，康樂，2010，10：107-116。
[3] BARBIERI F, SAGGION H. 推特中的反諷建模，2010，56-64。
[4] REYES C, BRACKETT M, RIVERS S, WHITE M, SALOVEY P. 課堂情緒氛圍, 學生參與度和學業成績。教育心理學雜誌，2012，104：700-712。
[5] BINSTED K, PAIN H, RITCHIE G. 兒童對計算機生成的雙關語的評價。語用認知，1997，5。
[6] PETROVIC S, MATTHEWS D. 從大數據中生成無監督的笑話。 訪問控制列表，2013。
[7] STOCK O，STRAPPARAVA C. 認真對待計算幽默的發展，2003，59-64。
[8] HOWARD J, RUDER S. 用於文本分類的微調語言模型。 相關性，2018。abs/1801.06146。
[9] DEVLIN J, CHANG M-W, LEE K, TOUTANOVA K. 伯特：用於語言理解的深度雙向轉換器的預訓練。arXiv 預印本，2018。arXiv：1810.04805。
[10] FAN A, LEWIS M, DAUPHIN YN。分層神經故事生成。 相關性，2018，abs/1805.04833。
[11] ATTARDO S. 幽默語言理論。沃爾特·德·格魯伊特 (沃尔特·德·格鲁伊特)，2010，1。
[12] RADFORD A, WU J, CHILD R, LUAN D, AMODEI D, SUTSKEVER I。語言模型是無監督的多任務學習器，2019。
[13] WOLF T, DEBUT L, SANH V, CHAUMOND J, DELANGUE C, MOI A, CISTAC P, RAULT T, LOUF R, FUNTOWICZ M, BREW J. 抱臉 的變形金剛：最先進的自然語言，2019.
[14] HOLTZMAN A, BUYS J, FORBES M, CHOI Y。神經文本退化的奇特案例。 相關性，2019，abs/1904.09751。
[15] KESKAR NS, MCCANN B, VARSHNEY LR, XIONG C, SOCHER R. 控制：可控生成的條件轉換器語言模型，2019。
[16] MUNEER A, FATI SM。使用深度學習基於形狀和紋理特徵的高效自動草藥分類方法。IEEE 訪問, 2020, 8: 196747-196764。
[17] DURAIRAJAH V, GOBEE S, MUNEER A. 使用 DNN 和 SVM 分類器的基於自動視覺的分類系統。 2018 第三屆控制,機器人和控制論國際會議（CRC），2018，6-14
[18] NASEER S, ALI RF, MUNEER A, FATI SM。 一世一種米德伏深的：使用深度學習和偽氨基酸組成預測蛋白質中的纈氨酸酰胺化位點。對稱性，2021，13（4）：560。
[19] NASEER S, ALI RF, FATI SM, MUNEER A。 一世硝基深：使用深度學習補充致癌研究的硝基酪氨酸位點的計算識別。 電子電氣設備 訪問，2021，9：73624-73640
[20] JOHNSON R, ZHANG T. 用於通過區域嵌入進行文本分類的半監督卷積神經網絡，2015。
[21] DE OLIVEIRA L, RODRIGO AL。 喊叫 評論中的幽默檢測，2015。
[22] VIG J. 變壓器模型中註意力的多尺度可視化。arXiv 預印本，2019。arXiv：1906.05714。
[23] CHEN P-Y, SOO V-W。使用深度學習的幽默識別。計算語言學協會北美分會 2018 年會議論文集：人類語言技術，2018，2：113-117。
[24] MUNEER A, FATI SM。推特上網絡欺凌檢測的機器學習技術比較分析。未來互聯網，2020，12（11）：187
[25] DE OLIVEIRA L, RODRIGO AL。 喊叫評論中的幽默檢測，2015。
[26] YANG Z, DAI Z, YANG Y, CARBONELL J, SALAKHUTDINOV R, LE QV。 網絡：語言理解的廣義自回歸預訓練，2019。
[27] RAFAY A,SULEMAN M,ALIM A. 基於機器和深度學習的魯棒評論評級預測模型：喊叫 數據集。2020 年智能技術新興趨勢國際會議 ，2020，8138-8143。