Investigate Strategies for Coming Up with Hilarious Text or Jokes Using NLP

Maliha Mussarrat, Abrar Qasem, Hasan Mohammod Tanzir,
Md. Abu Bakar Siddique Khan, Sadiul Arefin Rafi,
Md Sabbir Hossain, and Annajiat Alim Rasel
Dpt. of Computer Science and Engineering
School of Data Sciences (SDS)
BRAC University
Dhaka, Bangladesh
hasan.mohammod.tanzir@g.bracu.ac.bd
abu.bakar.siddique.khan@g.bracu.ac.bd

***Abstract*—In this thesis, we embark on an exploration of cutting-edge Natural Language Processing (NLP) techniques with the overarching goal of unraveling the enigmatic world of humor and wit. Our research revolves around delving into the fascinating realm of generating hilarious text and jokes through the powerful lens of NLP. Drawing upon the transformative advancements in NLP, we meticulously employ state-of-the-art language models, specifically leveraging the widely acclaimed GPT-2 transformer architecture. By meticulously fine-tuning and harnessing its capabilities, we uncover the hidden nuances of humor in textual content. To achieve our objectives, we propose a novel approach to extract amusement from NLP-generated sequences. Our method involves a meticulous selection of top-N tokens based on probability distributions, paving the way for crafting remarkably humorous output. Guided by the inherent creativity of the model, we adeptly curate punchlines and witty responses that evoke laughter. Amalgamating the merits of academic rigor with a touch of levity, our investigations traverse various dimensions of humor. We explore the impact of different token selection strategies and employ a randomization process to generate diverse, amusing outcomes. With a fervent focus on the ”human touch” in humor, our thesis culminates in the development of a tailored function for generating amusing text sequences. We meticulously examine its efficacy, ensuring that the generated content aligns with the principles of natural hilarity and wit. The implications of our research are far-reaching, as humor plays a quintessential role in human communication and social interactions. By unraveling the secrets of generating hilarious text through NLP, we aspire to augment a spectrum of applications, ranging from entertainment and creative writing to chatbot interactions and digital communication.**

***Index Terms*—NLP, GPT-2, Humor, Jokes, Text Generation, Creativity.**

I. INTRODUCTION

In the realm of Natural Language Processing (NLP), the quest for unlocking the subtleties underlying humor and wit has captivated researchers and practitioners alike. This academic inquiry delves into the nuanced art of generating humorous text and witty jokes through the advanced lens of NLP techniques. Guided by the transformative capabilities of the GPT-2 transformer architecture, our investigation seeks to elucidate the underlying mechanisms that evoke laughter and amusement within computational language models. At the core of our exploration lies the meticulous analysis of top-N token selection, facilitated by probability distributions, to curate humor-laden narratives. Our scholarly endeavor extends beyond superficial jests, immersing into the deeper psychological and linguistic underpinnings of humor. The aim is to discern the subtle nuances and linguistic constructs that contribute to the elicitation of laughter. Drawing from the wellspring of creativity, we meticulously craft punchlines and playful retorts, illustrating a keen understanding of the interplay between language and amusement. This academic pursuit transcends entertainment, reaching into the realms of interactive dialogue systems and human-computer interactions, where humor plays a vital role in enhancing engagement and fostering positive user experiences. By delving into the interstice of NLP and humor, we aspire to contribute substantially to the field of computational language modeling. Each witty utterance generated serves as a testament to the seamless fusion of technology and human ingenuity, unlocking new avenues for computational creativity. Beyond the realm of levity, our research elucidates the broader implications of humor, underscoring its capacity to foster genuine connections and augment the quality of human-machine interactions. We aspire to provide insights that resonate with diverse domains, ranging from creative writing to conversational agents, where the infusion of wit can elevate user satisfaction and cultivate memorable interactions. As we traverse this intellectual journey, our collective pursuit embraces the intricate interplay of linguistic artistry and technological advancement. This academic exploration beckons researchers, scholars, and practitioners to unite in our shared pursuit of unraveling the intricacies of humor within the digital landscape.

Identify applicable funding agency here. If none, delete this.

II. Literature Review

Theoretical aspects of humor and its connection to pun-related linguistic features have been examined in a number of studies. In order to divide puns into two dimensions—ambiguity of meaning and distinctiveness of viewpoints—Kao et al. (2016) presented a probabilistic model. They discovered that the intersection of these two factors significantly predicts how funny a pun is rated by people. He at 2019 expanded on this research by adding surprise as a new criterion for gauging the humor of puns. Their research shown that when words emerge unexpectedly in a local context yet make sense in a global context, surprise plays an important role in humor perception. Older methods of creating puns frequently depended on erroneous assumptions about semantic ambivalence. For instance, Yu et al. (2018) and Luo et al. (2019) used reinforcement learning and limited language models to encourage the ambivalence of pun phrases. These techniques, however, lacked solid theoretical underpinnings and had trouble producing clever and high-quality puns. Tian provided a unified framework for pun creation to address the shortcomings of earlier research and include the theoretical insights from computational comedy theories. Their three-part model, which consists of a context words/phrases picker, a label predictor, and a generation model, combines ambiguity, distinctiveness, and surprise. According to evaluation results, their unified model performed better at producing homophonic and homographic puns than strong baselines. Both homophonic and homographic pun creation have been covered independently in prior works. While homographic puns use words with numerous meanings that have the same spelling, homophonic puns use words that sound similar but have different meanings. By turning homographic puns to homophonic ones using word sense disambiguation, Tian bridges the gap between these two pun kinds. Their unified system can effectively produce both types of puns thanks to this move.Future research could refine substitute pun-alternative word pairs for homographic pun generation and overcome content biases.

The process of recognizing humor entails determining whether a sentence or conversation contains a certain level of humor. This endeavor is difficult because there are numerous types of humor, including irony, jokes, and wordplay. Numerous humour feature extractions have been proposed by researchers, including lexical and syntactic ambiguity, stylistic features such as alliteration and rhyming, and deep learning techniques. Recent deep learning techniques, such as Convolutional Neural Networks (CNN) and BERT-based methods, have acquired popularity for detecting humor . Before the introduction of LSTM (RNN) for humor production, the vast majority of systems relied on templates for joke generation. LSTM and Seq2Seq models have been used to generate humor, but it remains difficult to achieve human-like quip generation. The data set utilized in this study was compiled from numerous sources and measures. It includes two-sentence positive examples and encompasses not only structured jokes but also informal conversations. Due to a dearth of sufficient training data, LSTM and Seq2Seq models exhibited subpar performance. Even with a reduced dataset, the optimized GPT-J model demonstrated superior performance and excellent generalization capabilities. In terms of generating amusing responses, the fine-tuned GPT-J model outperformed LSTM and Seq2Seq models, according to the results. With a reduced dataset, the GPT-J model exhibited superior generalization. Based on conversational data, the study presented methods for detecting and generating humorous conversations. In comparison to conventional models such as LSTM and Seq2Seq, the GPT-J model exhibited superior performance. The pre-train and fine-tune technique showed promise in tasks requiring the generation of humor. Future work may include customizing the model for various conversational forms and collecting more diverse data from a variety of sources.

This paper introduces ”Witscript,” an innovative joke generator built for use in chatbots. The paper ”Witscript: A Novel Joke Generation System for Chatbots” presents Witscript, a novel joke generation system that takes into account the current conversational environment. The technology is meant to be included into chatbots, with the ultimate goal of making the chatbot more human and endearing through the use of humour. The study begins by describing the problems with current chatbot humour generators. In order to generate humour, many current systems either use prewritten jokes or generate jokes that stand alone and hence lack context. Witscript, on the other hand, is designed to mimic a witty human buddy by coming up with fresh jokes that fit within the context of the conversation. The Surprise Theory of Laughter, established by a comedy writer, serves as the theoretical foundation for the joke creation algorithms used in Witscript. Witscript uses this principle to generate incongruous one-liners by comparing the frequency of occurrence of two topic keywords. Two topic keywords from the user's input sentence are used as the foundation for the joke's topic handles, which are then used to generate the joke. The algorithm then makes an effort to build a pun out of these keywords through the use of wordplay. The juxtaposition, substitution, and portmanteau wordplay lines are discussed. The method determines which potential punch line has the highest wordplay score by comparing the quality of wordplay demonstrated by two provided words. Witscript creates an angle that naturally transitions from the user's input sentence to the joke's punch line, improving the joke's coherence and naturalness. Tokens are predicted to bridge the gap between the user's input and the punch line using a language model trained on jokes (Bidirectional Encoder Representations from Transformers). As part of the paper's literature assessment, the authors touch on previous research into computational humour systems. They note that many existing systems either use pre-made jokes or generate jokes that are completely disconnected from their surroundings. Conversely, Witscript fills the void of conversational humour that is both spontaneous and sensitive to context. In terms of producing responses that were deemed humorous by human evaluators, Witscript performed better than the baseline model

(DialoGPT). Compared to DialoGPT, whose responses were judged as humorous less than 20% of the time, Witscript's comments were deemed humorous more than 40% of the time.Overall, the paper introduces Witscript as a cutting-edge, very efficient method for producing fresh, topic-appropriate jokes for chatbots. Witscript's ability to generate humour that users find funny indicates a significant step in the direction of giving chatbots a more naturalistic sense of humour. In order to give consumers with engaging and likeable interactions, the authors highlight that Witscript might be used into open-domain chatbots. The potential of Witscript-enabled chatbots as entertaining companions for people in search of social connection is emphasised as the paper draws to a close.

The ”AmbiPun” paper introduces an innovative method for producing homographic puns by using context words. The authors present the argument that pun ambiguity is not due to the pun term itself but to the context in which it is used. This worldview is consistent with theories of humorous behaviour and serves as the basis for their proposed approach. The study begins by outlining the role of computational humour in NLP and highlighting the difficulty of pun generation caused by a lack of substantial training data. Unfortunately, existing methods are generally constrained by extensive training or mathematical models. The authors address these difficulties by suggesting a straightforward method for fixing the issue. They use a reversible dictionary to produce lexical matches for each of the pun's possible meanings. This process aids in separating the two meanings and enables the use of context terms that are monosemous for both meanings. Extractive (with TF-IDF), similarity (with Word2Vec), and generative (with GPT3) approaches are investigated to acquire context words. The similarity-based approach employs Word2Vec to locate context words based on word connections, whereas the extractive approach extracts keywords from retrieved sentences. The GPT3 is used in the generative technique to produce context words from a small number of samples. This study presents a new keyword-to-sentence model (T5) for creating pun-filled phrases from a pun term and its surrounding words. In addition, a humour classifier (BERT-large) is used to evaluate and rank candidate sentences for their level of hilarity and logical flow. The experimental outcomes prove that the proposed method outperforms the current standards. In both automatic and human evaluations using Amazon Mechanical Turk (AMT), the strategy improves success rates, hilarity scores, and sentence-level diversity. The benefits of the extractive method over the generative approach in terms of humour and pun success rates are discussed in the latter section of the study. It is also investigated where in the prompt the pun word should go; it is found that putting it near the conclusion of the sentence leads to more humorous puns. In conclusion, ”AmbiPun” provides an attractive approach to producing homographic puns that make use of surrounding terms to produce ambiguity and humour. The method outperforms the current state-of-the-art algorithms in pun generation because of its simplicity, efficiency, and originality. This study makes a significant contribution to the field of computational humour and paves the way for further investigation into the generation of hilarious sentences that are not limited by their semantic structure.

III. DATASET

To efficiently generate new humor using machine learning, models must understand the deep semantic intricacies of jokes. The intricacy stems from the need to comprehend the underlying humor, which frequently includes wordplay, incongruity, and surprise aspects. Solving such problems is challenging for a variety of reasons, one of which is the paucity of a comprehensive database containing a large number of jokes. To address this issue, a large dataset of over 0.2 million jokes was generated by crawling multiple websites that provide funny and short jokes.

A. Data Collection

The dataset is a csv file that contains 231,657 jokes which was acquired from Kaggle. The length of the jokes in the dataset ranges from succinct 10-character jokes to more complicated ones with up to 200 characters. This variation in length guarantees that the dataset covers a wide range of comedy styles, catering to various preferences and sorts of jokes.

B. Tokenization

A tokenizer object made specifically for the 'gpt2-medium' variation is produced using the GPT2Tokenizer method. Tokenizers are in charge of dividing raw text into tokens, which are smaller units that can be entered into the GPT-2 model. The GPT-2 language model is then initialized using pre-trained weights from the 'gpt2-medium' variation using GPT2LMHeadModel method. This variation corresponds to a medium-sized GPT-2 model, which balances model capacity and computational resources. The GPT-2 model is a particular transformer-based language model that could predict the following word in a sequence depending on the prior context.

C. PyTorch

PyTorch is a commonly used framework for deep learning research and applications because it offers effective tensor computations with automatic differentiation capabilities. A ”tensor,” a multidimensional array with GPU acceleration equivalent to NumPy arrays, is the basic data structure in PyTorch. Additionally, PyTorch offers a number of operations to effectively manipulate and process these tensors. Torch.utils.data.DataLoader and torch.utils.data are the two data primitives that PyTorch offers.Datasets that enable you to use both your own data and preloaded datasets. DataLoader wraps an iterable around the Dataset to make it easy to get the samples, which Dataset uses to store the samples and their related labels.

ACKNOWLEDGMENT

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g". Avoid the stilted expression "one of us (R. B. G.) thanks ...". Instead, try "R. B. G. thanks...". Put sponsor acknowledgments in the unnumbered footnote on the first page.

D. LaTeX-Specific Advice

Please use "soft" (e.g., `\eqref{Eq}`) cross references instead of "hard" references (e.g., `(1)`). That will make it possible to combine sections, add equations, or change the order of figures or citations without having to go through the file line by line.

Please don't use the `{eqnarray}` equation environment. Use `{align}` or `{IEEEeqnarray}` instead. The `{eqnarray}` environment leaves unsightly spaces around relation symbols.

Please note that the `{subequations}` environment in LaTeX will increment the main equation counter even when there are no equation numbers displayed. If you forget that, you might write an article in which the equation numbers skip from (17) to (20), causing the copy editors to wonder if you've discovered a new method of counting.

BibTeX does not work by magic. It doesn't get the bibliographic data from thin air but from .bib files. If you use BibTeX to produce a bibliography you must send the .bib files.

LaTeX can't read your mind. If you assign the same label to a subsubsection and a table, you might find that Table I has been cross referenced as Table IV-B3.

LaTeX does not have precognitive abilities. If you put a `\label` command before the command that updates the counter it's supposed to be using, the label will pick up the last counter to be cross referenced instead. In particular, a `\label` command should not go before the caption of a figure or a table.

Do not use `\nonumber` inside the `{array}` environment. It will not stop equation numbers inside `{array}` (there won't be any anyway) and it might stop a wanted equation number in the surrounding equation.

E. Some Common Mistakes

- The word "data" is plural, not singular.
- The subscript for the permeability of vacuum μ_0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter "o".
- In American English, commas, semicolons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
- A graph within a graph is an "inset", not an "insert". The word alternatively is preferred to the word "alternately" (unless you really mean something that alternates).
- Do not use the word "essentially" to mean "approximately" or "effectively".
- In your paper title, if the words "that uses" can accurately replace the word "using", capitalize the "u"; if not, keep using lower-cased.
- Be aware of the different meanings of the homophones "affect" and "effect", "complement" and "compliment", "discreet" and "discrete", "principal" and "principle".
- Do not confuse "imply" and "infer".
- The prefix "non" is not a word; it should be joined to the word it modifies, usually without a hyphen.
- There is no period after the "et" in the Latin abbreviation "et al.".
- The abbreviation "i.e." means "that is", and the abbreviation "e.g." means "for example".

An excellent style manual for science writers is [9].

F. Authors and Affiliations

The class file is designed for, but not limited to, six authors. A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

G. Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is "Heading 5". Use "figure caption" for your Figure captions, and "table head" for your table title. Run-in heads, such as "Abstract", will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced.

H. Figures and Tables

a) Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation "Fig. 1", even at the beginning of a sentence.

TABLE I
TABLE TYPE STYLES

Table Head	**Table Column Head**		
	Table column subhead	***Subhead***	***Subhead***
copy	More table copy[a]		

[a]Sample of a Table footnote.

Fig. 1. Example of a figure caption.

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity "Magnetization", or "Magnetization, M", not just "M". If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write "Magnetization (A/m)" or "Magnetization {A[m(1)]}", not just "A/m". Do not label axes with a ratio of quantities and units. For example, write "Temperature (K)", not "Temperature/K".

REFERENCES

Please number citations consecutively within brackets [3]. The sentence punctuation follows the bracket [4]. Refer simply to the reference number, as in [5]—do not use "Ref. [5]" or "reference [5]" except at the beginning of a sentence: "Reference [5] was the first . . ."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [6]. Papers that have been accepted for publication should be cited as "in press" [7]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [8].

REFERENCES

[1] A. MoudGil, "Short Jokes," Kaggle, 2017. [Online]. Available.

[2] R. Tanwar, "Cracking Jokes Using GPT2," Kaggle, 2020. [Online]. Available.

[3] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[4] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[5] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[6] K. Elissa, "Title of paper if known," unpublished.

[7] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[8] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[9] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[10] Mittal, A. (2022, May 4). AmbiPun: Generating Humorous Puns with Ambiguous Context. arXiv.org.

[11] Bertero, D. (2019). Conversational humor recognition and generation through deep learning. HKUST SPD — the Institutional Repository.

[12] Toplyn, J. (2023, February 3). WitScript: a system for generating improvised jokes in a conversation. arXiv.org.

[13] Mittal, A. (2022b, May 4). AmbiPun: Generating Humorous Puns with Ambiguous Context. arXiv.org.