

Supervised Text Classification: A Framework for AI-Generated Text

Anushtha Vishwakarma¹ and Dr. Mitali chugh²

¹⁻²UPES/Computer Science, Dehradun, India

Email: vishwakarmaanushtha@gmail.com, mchugh@ddn.upes.ac.in

Abstract— Prophet Net and Transformer models have both made significant advancements in the domain of text generation toward producing language that is comparable to human-authored content. For text generation on several platforms, including English, Chinese, and others, Prophet Net is an effective approach. A branch of natural language processing (NLP) termed text creation combines computational linguistics and artificial intelligence to create original material. A sophisticated deep learning model Bert, GPT models are excellent for expressing the idea and following fundamental grammatical norms. In our work, the method we suggested for automatically producing new text includes a language model that, given input data, generates new text pertinent to that data's domain. The fundamental basis for the idea of text generation is the Synthetic text Generator structure, which is comparable to Prophet Net and GPT. The generated data is cleansed in order to determine whether the text was produced by a machine or by a human author. The results of this study show the proportion of text that was categorized as human and machine-generated text produced by machine learning models.

Index Terms— NLP, Prophet Net, GPT-2, Logistic Regression, SVM, and Random Forest.

I. INTRODUCTION

There are many potential uses for text creation technology, which is an important area of study in the field of natural language processing. Intelligent question-and-answer systems, machine translation, and other applications can all benefit from the use of text-generation technology to produce more intelligent and natural human-computer interactions [1]. Sentence-level text creation makes an effort to carefully analyse the entire text and determine how each sentence relates to the context in which it is placed. Recent improvements in processing power and deep learning algorithms made it possible to produce texts automatically. [2]. Artificial neural networks and representation learning are both used in the machine learning discipline of deep learning. Microsoft researchers presented a neural language generation model called Prophet Net in 2020. The prominent Transformer model, which is renowned for its capacity to manage long-range dependencies in language modelling tasks, serves as the foundation for this model's architecture. Prophet Net is trained to anticipate word sequences that may appear several positions ahead in the text, in contrast to typical language models that only forecast the next word in a sentence. As a result, the model can better capture long-term dependencies and produce higher-quality content. Transformer models create text using their neural network design capabilities to identify word associations and patterns in massive textual corpora. They specifically employ a technique known as "sequence-to-sequence" learning (Seq2Seq), which entails training the model to accept a sequence of input text (such as a prompt or a sentence) and produce a sequence of output text (such as a completed sentence or paragraph). By generating text

this way, the transformer model is able to produce highly coherent and fluent language that is indistinguishable from text written by humans.

II. LITERATURE SURVEY

Natural Language processing in text generation is an important area of research in the era of science. Transformer pre-trained model yet not produce optimal results. RNN and LSTM model performs well in the generation of a long sequence of text. [3] The gated loop unit (GRU) and the LSTM with a peephole connection are two examples of improved LSTM variants for various jobs, all of which perform better than the LSTM in those activities.

The paper proposed a novel pre-training objective called future n-gram prediction, which is designed to help the model capture longer-term dependencies in the text. Specifically, instead of predicting the next token in the sequence, the model is trained to predict the future n-grams in the text, where n can be a variable length. This allows the model to learn more complex relationships between the tokens and improve its ability to generate coherent and contextually appropriate [4] text.

One of the primary goals of automated text categorization is to develop algorithms that can classify text documents into predefined categories based on their content. This has many practical applications, such as classifying news articles, filtering spam emails, and organizing large collections of digital documents. One common approach in automated text categorization is to use a bag-of-words representation of the text data, where each document is represented as a vector of word frequencies. This approach can be combined with various machine-learning algorithms to build a text classifier. Deep learning techniques such as CNN and RNN can also improve classification accuracy. Ensemble learning methods can further improve the performance of text [5] classifiers.

The paper describes the development of a synthetic electronic medical record (EMR) text generation system that can generate realistic EMR text. Here authors discuss the importance of EMR data for healthcare research and the challenges associated with accessing and using such data. They note that one solution to this problem is to generate synthetic EMR data that can be used for research without compromising patient privacy. The proposed system uses a combination of rule-based and deep-learning techniques to generate synthetic EMR text. The rule-based approach generates syntactically correct sentences, while the deep learning approach generates semantically correct sentences. The system is trained on a large corpus of real EMR data and is evaluated using various metrics such as perplexity, BLEU score, and human [6] evaluation.

The study discusses the importance of synthetic text data in various applications, such as natural language processing and machine learning. They also highlight the privacy concerns associated with using real text data, which may contain sensitive information. To address these privacy concerns, here the authors propose a method for generating synthetic text data that provides differential privacy guarantees. The method involves training a language model on real text data and then using the model to generate synthetic text data with added noise. The authors demonstrate that their approach provides strong differential privacy guarantees, but it is unclear how much utility is preserved in the synthetic text [7] data.

In this paper, the author surveyed factors for fine-tuning of the pre-trained language model. Which first weight initialization includes different subsets of the pre-trained weights. Then Order of the data which examines in random order, sorted by length, and sorted by label and they find that sorting the data by the label can improve performance, especially for low-resource datasets. The paper provides a useful overview of the impact of these key factors on fine-tuning pre-trained language models and highlights the importance of carefully considering these factors when applying fine-tuning to new [8] tasks.

The paper presents an empirical study on detecting fraudulent hotel reviews using a combination of human and machine intelligence. The usefulness of four approaches (Naive Bayes, SVMs, human computation, and hybrid human-machine approaches) in separating real reviews from false ones is tested by the authors by creating fake reviews and merging them with verified reviews. The results demonstrate that assuming the most important features are selected, a hybrid human-machine strategy outperforms either human or machine-based methods. Overall, the paper suggests that a combination of human and machine intelligence is a promising approach for detecting fraudulent reviews and has broader implications for fraud detection in other [9] domains.

Recent research has demonstrated that GPT-2 produces excellent text in a range of fields, including news. One of these studies showed that GPT-2 is capable of producing news articles with a high level of fluency and coherence. Based on this work, we generate news-like articles for our analysis using GPT-2 [10].

GPT-2 makes it possible to generate text intuitively by leveraging the previous context to estimate a probability distribution over the vocabulary of the model, which can then be simply decoded by selecting the following word from it. Besides the new architecture, these language models profit mainly, from training over bigger datasets. [11]

As can perform well across multiple domains and datasets when it is trained on a sufficiently large and diversified dataset.

The paper proposes a method for automatically generating natural language descriptions for code blocks using a combination of static analysis and machine learning techniques. The authors present a system that extracts relevant information from code blocks, such as variable names and data types, and uses this information to generate natural language comments. The authors evaluate their system on a dataset of code snippets and compare it with a baseline system that simply repeats the code as comments. The results show that their system outperforms the baseline system in terms of both relevance and readability of the generated comments. The paper suggests that the automatic generation of natural language comments for code blocks can improve the understandability and maintainability of software systems, and has implications for software engineering and programming [12] education.

The paper presents a unified text-to-text transformer architecture called T5, which is trained on a diverse set of natural language processing (NLP) tasks using a single objective function. The authors explore the limits of transfer learning with T5 by evaluating its performance on a range of downstream tasks, including language modelling, question answering, and text classification. The authors find that T5 outperforms existing state-of-the-art models on several NLP tasks, demonstrating the effectiveness of the unified text-to-text approach to transfer learning. Overall, the paper suggests that the unified text-to-text transformer architecture has significant potential for improving the performance of transfer learning in [13] NLP.

III. METHODOLOGY

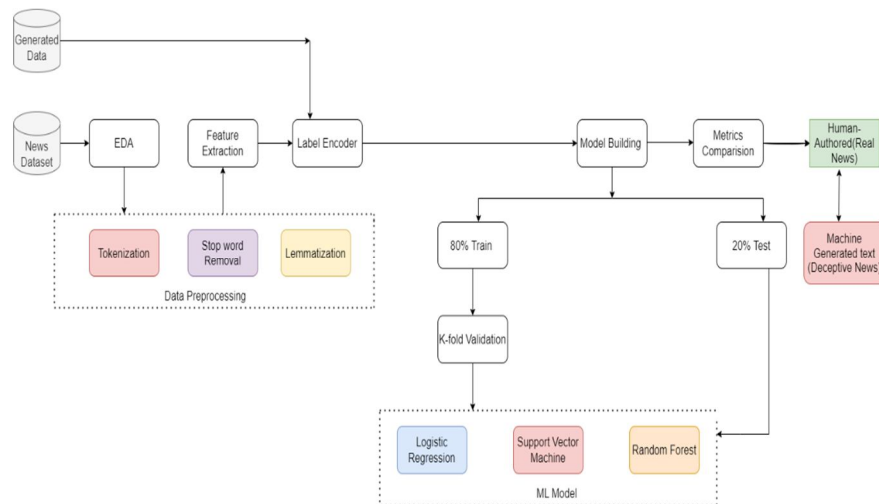


Figure 1. Proposed work

A. Data Pre-processing

Here in our Fig.1 proposed work, we are using two different datasets the first is a human-generated dataset that we have downloaded from Kaggle which is verified by PolitiFact.com that contains id, title, author, label, and text. The second, dataset is from the GitHub repository which we have used in our work for both generative models. We have taken 1000 rows of data from each dataset. The generative models contain 1000 rows as input data which is explained further in the next step. Before passing on the input text for a generation we removed all the noise from the data to get rid of all the unwanted problems for further analysis. The NLTK library is used for removing stop words, the lower casing of text, removing unwanted characters such as links and numeric integers, etc. After that data is cleaned and used further for text generation.

B. Text Generation

For text generation, we are using Prophet Net and GPT-2 tools. Prophet Net works by learning to predict the next word or sequence of words in a given input sequence of text. It uses a series of attention mechanisms to identify important words and phrases in the input sequence, which it then uses to generate the next word or sequence of words. By using Prophet net pre-trained libraries such as Prophet Net Tokenizer, and Prophet Net For Conditional Generation. from_ pretrained the synthetic text is generated.

GPT-2 works by training on a massive corpus of text data, such as web pages or books. It then uses this learned knowledge to generate text content in response to a given prompt. The model generates text by predicting the next

word or sequence of words in the input sequence, based on the context of the prompt and its own internal representations of language patterns and structures.

Here by using these two tools the synthetic text is generated and labelled as machine-generated text and further by using a machine learning algorithm classification of text between human-authored text and machine-generated text. In which the evaluation metrics are used to evaluate textual data. And the evaluated result describes which generative model output results(text) need to take further for model building.

C. Text Data Model Evaluation

1. Bleu Score

Here we evaluate our textual data on Bleu Score. The BLEU score is then calculated as the weighted geometric mean of the individual precisions, where the weights are based on the length of the n-grams. The BLEU score ranges from 0 to 1, with higher scores indicating a better-quality machine-generated text.

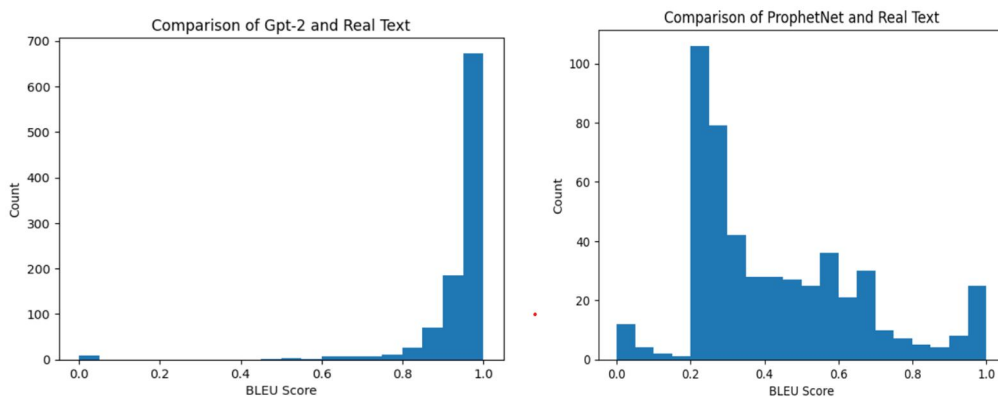


Figure 2. Bleu score evaluation of text generation

As seen in Fig. 2 in GPT-2 most of the data were in the range between 0.6 to 1.0 this score indicates better quality of the machine-generated text. And in Prophet Net, most of the data were in the range of 0.2 to 0.8. As a generated text of GPT-2 is almost touching the value one it gives the conclusion that the generated text has good relativity to the input text provided. so, we proceed to work with GPT-2.

D. Data Pre-Processing and EDA on the Generated Text

After the generation of textual data, pre-processing and EDA on generated text, because there might stop words, noisy data, or unwanted characters, can be there so to remove unwanted data and special character URL, non-ASCII characters white space, etc pre-processing needs to perform. To use this text for further analysis we are performing pre-processing so that it doesn't give any obstacles while performing analysis and word cloud.

To get some insights and information from the dataset, Exploratory data analysis [EDA] describes the shape of the overall data frame, and the type of data such as an object, integer, etc. Checking for duplicate values and unique values and then performing visualization concerning news made by people. In this work, EDA performed graphically well to have some good insights which help in knowing the process of generated text as well as the relative ness between the given input and generated output text. Like method as location.

E. Grouping

Here the two different datasets have been merged under a column as a group which contains two different labeling named Human-authored text and Machine-generated text. Since it will become easier for machines to classify these two texts so we have shuffled the text.

```
shuffled = new_data.sample(frac=1, random_state=1).reset_index()
```

Making it more complicated for the machine to classify the text which is used further while performing the model building.

F. Tokenization

This is the most important step while performing text mining as it breaks every sentence that is making an individual list of each word of each sentence, which we commonly call tokens. The stop words list from NLTK

was downloaded, which helps in removing all the unwanted words. We are performing tokenization, cleaning the lists from stop words, and finally creating a list of lists into a single list of words. Here we import the library counter which counts the most common words for both Human-authored text and machine-generated text.

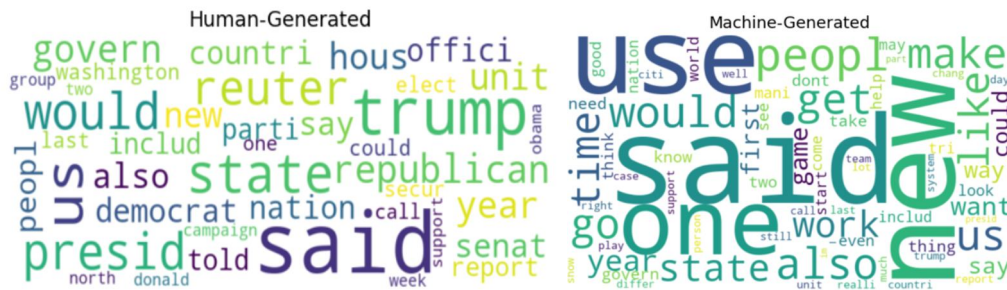


Figure 3. Most common words in Human-authored text and Machine-generated Text

Fig. 3 depicts that the AI-generated has a considerable number of similar words present in it that match the human-authored text. Like ‘trump’, ‘govern’, and ‘nation’. From this, with the growth of technology, bots can match up with people’s sentiments while writings articles, reviews, and news. Since AI-based models are taught to mimic human language, it is difficult for a person to differentiate between human-authored and AI-generated text.

G. Label Encoder

Before performing the label encoder, we grouped our data into two classes human-authored text and machine-generated text.

```
new_1["group"] = " human-authored "
```

```
df["group"] = " machine-generated text "
```

Then applying label encoder on the group which works on categorical data and is used for the normalization of text. It formats labels into machine readable form where it converts labels into integer format. It is the most important step for applying supervised learning to structured data. By importing the SK learn library which is a tool for label encoding.

H. Vectorization: TF-IDF

It involves calculating the frequency for each term inside the corpus, which means the IDF scores of each word in our corpus. It shows the significance of words in the document and the whole corpus, as the word importance denotes information fetched from the data. A bag of words only converts the text into a feature vector; it does not consider the words' importance. In this study, we divided our whole data into an 80:20 ratio. The remaining 20% of the data are used to test the performance of the models we have selected, while the remaining 80% of the data are used to train the models.

I. Algorithm

Input: News Dataset

output: Detection and classification of News text between Machine_generated text and human_authored_text

Split the dataset into two parts: one containing machine-generated text and the other containing human-authored text.

upload dataset and preprocess

Text Generation using Prophet net & GPT2

Perform Text Evaluation using Blue Score

Identify the best Text generation Tool on the basics of Bleu Score

Preprocess Generated Text

Perform EDA for feature extraction:

```
shuffled = new_data.sample(frac=1, random_state=1).reset_index()
```

```
new_1["group"] = "human-authored"
```

```
df["group"] = " machine-generated text "
```

Perform tokenization and stop word removal

Calculate tf_idf:

$tf - idf = tf * idf$
 where $tf-idf$ = Term frequency-inverse document frequency,
 tf = Term Frequency,
 idf = Inverse Document Frequency,
 t = Term, and d = Document.

Build the model using the following classification methods

Logistic Regression
 Support Vector Machine(SVM)
 Random Forest

Text Classification as human-authored text/machine-generated text.

J. Classification Model

1. Logistic Regression

Logistic regression is a statistical method used to analyze data and make predictions based on the relationships between one or more independent variables (also called predictors or features) and a binary or dichotomous dependent variable (also called the outcome or response variable). The method is called "logistic" because it models the logarithm of the odds (or logit) of the dependent variable taking on a certain value (e.g., 1 or 0).

2. Support Vector Machine

A supervised learning approach called Support Vector Machine (SVM) is employed for classification and regression analysis. SVM is a discriminative classifier that searches for the most effective decision boundary or hyperplane to categorize the input. SVM seeks to identify the hyperplane that maximizes the margin between the two classes in the case of binary classification.

3. Random Forest

Both classification and regression tasks are accomplished using the supervised machine learning method known as Random Forest. Using an ensemble learning method that incorporates various decision trees, a final prediction is created. By choosing a random subset of characteristics and a random portion of training samples for each tree, Random Forest builds a collection of decision trees from the training data.

IV. RESULTS

A. Performance Metrics

Performance measures that allow us to evaluate how well the model is performing are necessary in order to calculate or quantify the model's quality. We evaluate our model on Accuracy metrics. A classification report is used to assess the precision of predictions provided by a classification algorithm. To be more specific, True Positives, False Positives, True Negatives, and False Negatives are used to forecast the categorization metrics.

1. Accuracy

Accuracy evaluation is one of the most frequently used. It provides information on the proportion of predictions generated using a given machine learning model that was accurate. The model is strong if its accuracy value is high.

2. Model Comparison

TABLE I. MODEL RESULTS COMPARISON

Model Name	Test Accuracy	Train Accuracy
Logistic Regression	0.91	0.97
Support Vector Machine	0.83	0.87
Random Forest	0.94	0.96

The various machine-learning algorithms listed in Table 1 have been applied here. Logistic Regression, Support Vector Machine (SVM), and Random Forest, for our baseline accuracy forecast, we employed a classification model called logistic regression. We decided we needed to use more classification algorithms when the logistic regression's performance was adequate. The support vector machine, one of the top machine learning algorithms for categorizing data, was the next method employed. This model performed well when it came to text classification. Random Forest, the third algorithm employed, was valuable for our work since it had decision-making capabilities comparable to those of humans. Similar to the other algorithms, it delivered a strong

performance. To distinguish between human- and machine-generated text in this case, where the Random Forest machine learning model, offers good accuracy of 0.94 percent.

V. DISCUSSION

The recent advancements in text generation models, such as Prophet Net and Transformer models, have demonstrated remarkable progress in generating human-like language across different languages and platforms. The field of natural language processing (NLP) has been actively working on text creation, which combines computational linguistics and artificial intelligence to generate original content. In this context, the method proposed in this discussion, which involves a language model that generates new text based on input data, can be an effective approach for creating new content. The proposed concept of text generation is based on the framework of Synthetic Text Generator, which is similar to Prophet Net and GPT. Overall, the proposed approach can be a valuable tool for generating original content, particularly in content creation, marketing, and advertising industries. However, it is essential to ensure that the generated content is of high quality and is not misleading or harmful to the users.

VI. CONCLUSION

People are dependent on modern methods to make their lives easier as technology advances. In the age of automated systems, it is simpler to produce false information, and the likelihood of doing so is increased. So here we are generating synthetic text using generative models for increasing machine complexity. For that first, we perform data pre-processing and EDA where removal of unwanted text, links, etc. As we evaluate our model performance on performance metrics that are bleu score where it identifies the similarity between given input and generated output texts. With the help of these generative models, the generation of text is easier for people. As many industries use tools for the creation of a large amount of high-quality text like human-written text in a short amount of time. For the detection of human-authored text and machine-generated text here we used machine-learning models. Random forests perform well compared to other models.

REFERENCES

- [1] S. Lu, Y. Zhu, W. Zhang, J. Wang, and Y. Yu, "Neural Text Generation: Past, Present and Beyond," Mar. 2018.
- [2] N. Fatima, A. S. Imran, Z. Kastrati, S. M. Daudpota, and A. Soomro, "A Systematic Literature Review on Text Generation Using Deep Neural Network Models," *IEEE Access*, vol. 10, pp. 53490–53503, 2022, doi: 10.1109/ACCESS.2022.3174108.
- [3] L. Li and T. Zhang, "Research on Text Generation Based on LSTM," *International Core Journal of Engineering*, vol. 7, p. 2021, doi: 10.6919/ICJE.202105_7(5).0067.
- [4] W. Qi et al., "ProphetNet: Predicting Future N-gram for Sequence-to-Sequence Pre-training," Jan. 2020, [Online]. Available: <http://arxiv.org/abs/2001.04063>
- [5] F. Sebastiani, "Machine learning in automated text categorization," *ACM Comput Surv*, vol. 34, no. 1, pp. 1–47, Mar. 2002, doi: 10.1145/505282.505283.
- [6] J. Guan, R. Li, S. Yu, and X. Zhang, "Generation of Synthetic Electronic Medical Record Text," in 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), IEEE, Dec. 2018, pp. 374–380. doi: 10.1109/BIBM.2018.8621223.
- [7] X. Yue et al., "Synthetic Text Generation with Differential Privacy: A Simple and Practical Recipe," Oct. 2022, [Online]. Available: <http://arxiv.org/abs/2210.14348>
- [8] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, and N. Smith, "Fine-Tuning Pretrained Language Models: Weight Initializations, Data Orders, and Early Stopping," Feb. 2020, [Online]. Available: <http://arxiv.org/abs/2002.06305>
- [9] C. G. Harris, "Comparing Human Computation, Machine, and Hybrid Methods for Detecting Hotel Review Spam," 2019, pp. 75–86. doi: 10.1007/978-3-030-29374-1_7.
- [10] J. B. L. D. M. F. Y. C. Ari Holtzman, "The Curious Case of Neural Text Degeneration," Cornell University, 2020.
- [11] W. J. C. R. L. D. A. D. S. I. Radford A, "Language models are unsupervised multitask learners," 2019.
- [12] Y. Liang and K. Zhu, "Automatic Generation of Text Descriptive Comments for Code Blocks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018, doi: 10.1609/aaai.v32i1.11963.
- [13] C. Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," 2020. [Online]. Available: <http://jmlr.org/papers/v21/20-074.html>.