



Research Intern at IIITV

BTP Presentation On:

Application of Natural Language Processing for Requirements Engineering

Name: Abhiyank Raj Tiwari

Student Id: 201951011

Computer Science and Engineering

Under Supervision of : Dr. Novarun Deb



CONTENT OF THE PRESENTATION

- ❖ Brief Introduction of the Projects
- ❖ Text Summarization and Text Classification
- ❖ Identifying Paraphrased Sentences As Human Written OR AI Generated
- ❖ Paraphrasing With T5 Model
- ❖ Perplexity Calculation With GPT2 Model
- ❖ Calculating Log-Likelihood using GPT2
- ❖ Analysis on Log-Likelihood using GPT2
- ❖ Future Works
- ❖ Key Learning



Introduction

- Natural Language Processing (NLP) has various applications such as machine translation, sentiment analysis, and text summarization.
- Language models like T5 can paraphrase text at a paragraph level, producing lengthier and higher quality paraphrases.
- GPT2 can be used to calculate the perplexity of sentences for classifying them as human-written or AI-generated



Text Summarization and Text Classification

- ❖ Preprocessing of data:
 - Cleaning the text data by removing irrelevant information, such as punctuation, and stop words.
 - Normalizing the text by converting it to lowercase, stemming or lemmatizing the words.
 - Tokenizing the text into words or subwords.
- ❖ Model training:
 - Selecting an appropriate machine learning algorithm, such as Naive Bayes, SVM, or neural networks.
 - Splitting the labeled data into training and validation sets to evaluate the model's performance..
- ❖ Model evaluation:
 - Evaluating the model's performance on a held-out test dataset, using metrics such as accuracy, precision, recall, and F1-score.
 - Interpreting the model's results to gain insights into the data and improve the model's performance.

```
print("Word count of the original paper: ", len(paper_1))  
print("Word count after summerization: ", len(summary))
```

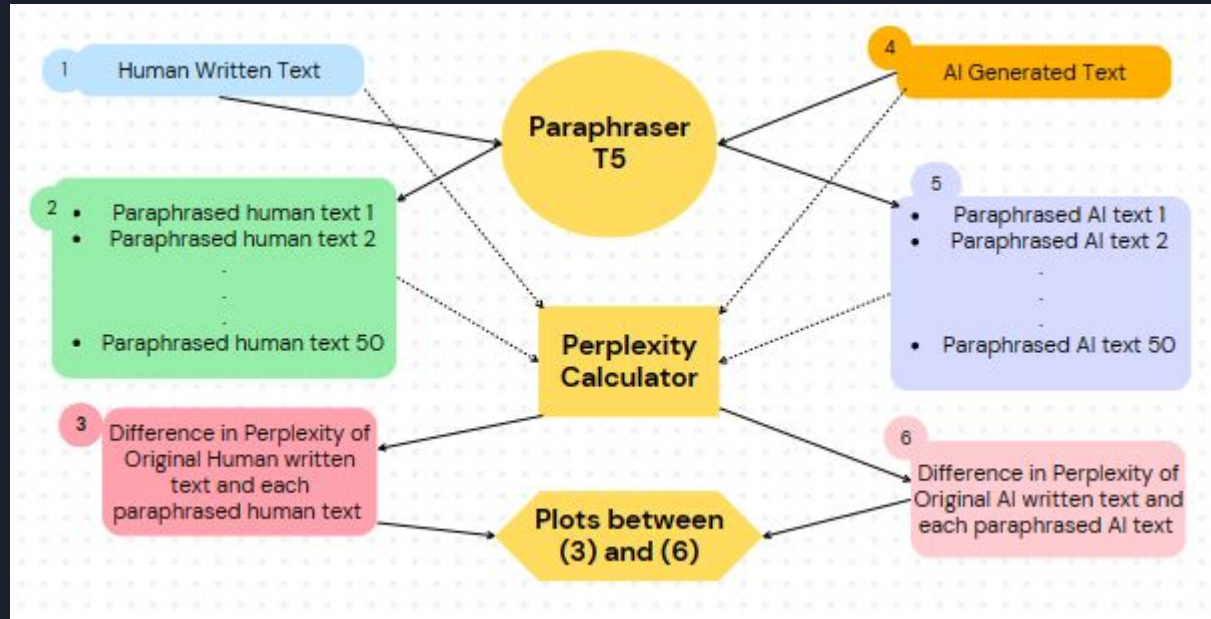
```
Word count of the original paper: 32749  
Word count after summerization: 11636
```

Output of text Summarization

```
array([[368, 62],  
       [133, 82]])
```

Confusion Matrix of text Classification


Identifying Paraphrased Sentences As Human Written OR AI Generated






Paraphrasing With T5 Model:


- The T5 model was used for paraphrasing text.
- The pre-trained T5 model was fine-tuned on a paraphrasing dataset.
- The dataset consisted of pairs of original and paraphrased sentences.
- The model was trained to generate a paraphrase for a given input sentence.
- The T5 model was observed to generate paraphrases that were longer and of higher quality than those generated by previous models.



```
prefix = "paraphrase"
pred1 = trained_model.predict([f""{prefix}: Calling a function or as
Dynamic binding can be associated with run time polymorphism and inhe
Dynamic binding makes the execution of a program flexible as it can d
and which function should be called, at the time of program execution
But as this information is provided at run time it makes the executio
slower as compared to static binding.""'])
df1 = pd.DataFrame(pred1).T
df1.columns = ["sentences"]
df1.to_csv("DynamicBinding_Human_T5_paraphrased.csv", index=False)
```

Input paragraph for T5 paraphrasing model

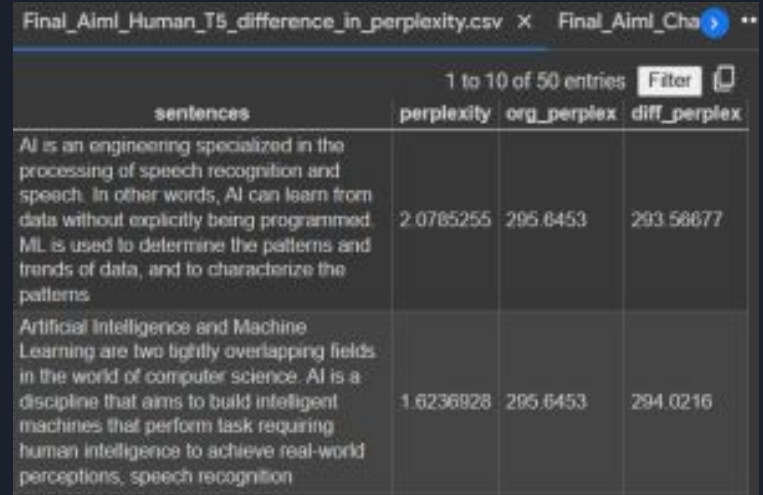


dynamicBinding_Human_T5_paraphrased.csv		DynamicBinding_ChatGPT_T5_>	
		1 to 10 of 50 entries	Filter 
sentences			
Dynamic binding is a technique that can resolve references to variables and other symbols at runtime. In dynamic binding, the mapping between the reference and the actual memory location is not determined until the program is executed. The binding between the method name			
The method for implementing a dynamization is a method to fix errors of objects, functions and other symbols at runtime. Dynamization is a technique used by computer programming languages to resolve the error or the error within			
Dynamic binding is the mechanism used in computer programming languages to resolve any such data at runtime. Such binding uses are intended to generate the information into its source by its source.			
Dynamic binding is a system used in programming languages to resolve reference to variables, functions and other symbols at runtime. Dynamic binding is not required in real-time programming languages, but in real time programming languages, where objects can have			
Dynamic binding is a tool used to create a reference to variables and functions at runtime. While the mapping of the reference-point is not until the program is executed, the results can also be determined by one method executing the			
Dynamic binding is a mechanism used in programming languages to resolve objects, functions, and other symbols at runtime. Unlike dynamic binding, the mapping of reference memory takes place until the program is executed. Using Dynamic binding, it			

Output Generated by T5 model

Perplexity Calculation With GPT2 Model

- Perplexity is a measure of how well a language model predicts a given sequence of words.
- The pre-trained GPT2 model was fine-tuned on a dataset of human-written and AI-generated sentences.
- The perplexity of a sentence was calculated by feeding it to the GPT2 model and computing the model's probability distribution over the next word.

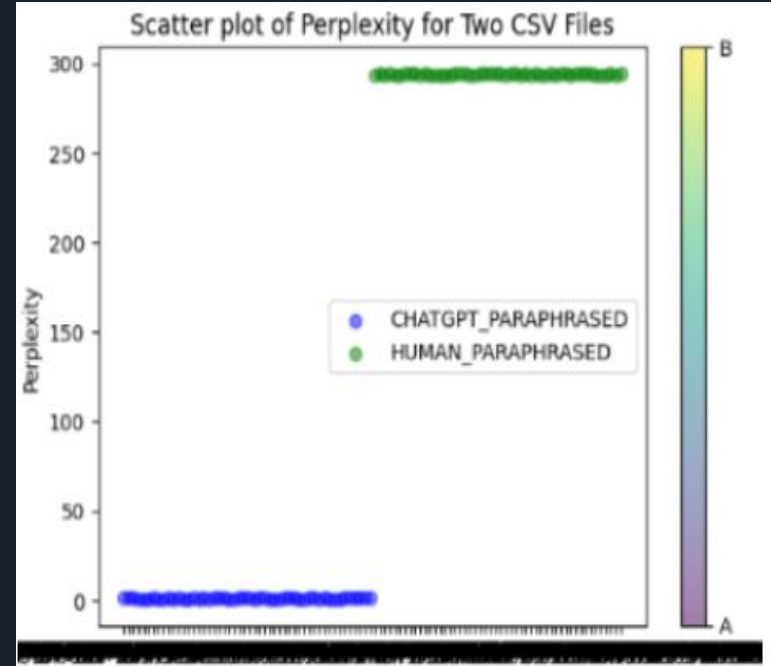


The screenshot shows a CSV file named 'Final_Aiml_Human_T5_difference_in_perplexity.csv' with 50 entries. The table displays the following data:

sentences	perplexity	org_perplex	diff_perplex
AI is an engineering specialized in the processing of speech recognition and speech. In other words, AI can learn from data without explicitly being programmed. ML is used to determine the patterns and trends of data, and to characterize the patterns	2.0785255	295.6453	293.56677
Artificial Intelligence and Machine Learning are two tightly overlapping fields in the world of computer science. AI is a discipline that aims to build intelligent machines that perform task requiring human intelligence to achieve real-world perceptions, speech recognition	1.6236828	295.6453	294.0216

Discussion on Perplexity Graph

- In an ideal case, the plot of perplexity scores for human-written and AI-generated sentences would show a clear separation between the two categories.
- The limitations of the transformer models themselves, such as the inability to understand context beyond a certain scope, can further affect the results





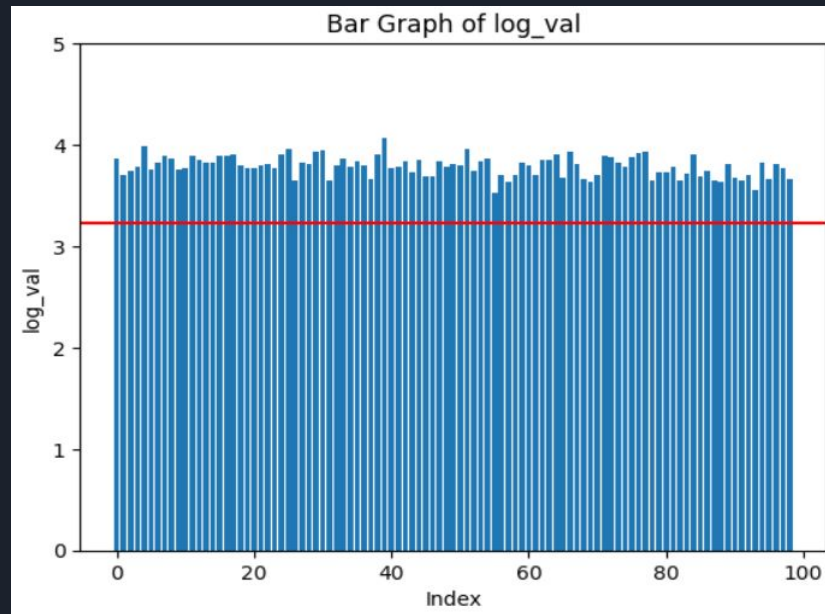
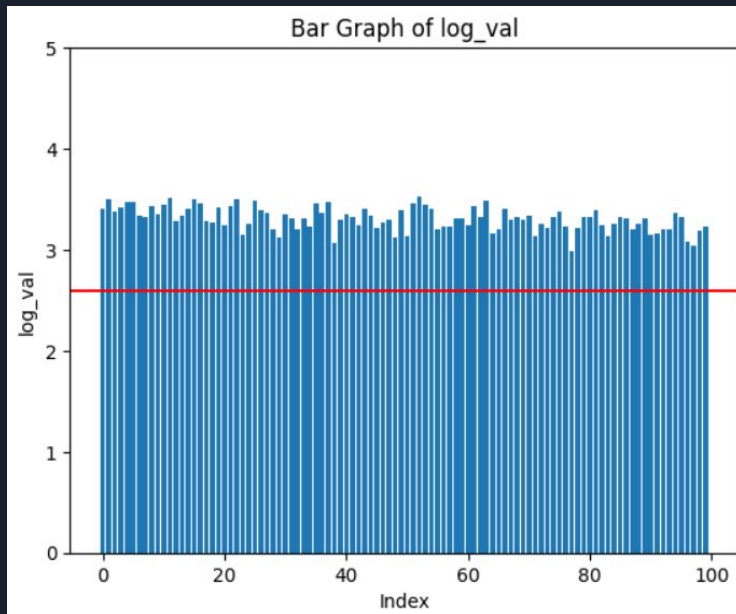
Calculating Log-Likelihood using GPT2

- Log-likelihood is a measure of how probable a sequence of words is according to a language model.
- The log-likelihood of a sentence is calculated by feeding it to the GPT2 model and computing the model's log probability distribution over the next word.
- The log-likelihood score is the sum of the log probabilities of each word in the sentence.
- Like perplexity, log-likelihood can also be used to distinguish between human-written and AI-generated sentences, with lower log-likelihood scores indicating a higher likelihood of the sentence being generated by AI.

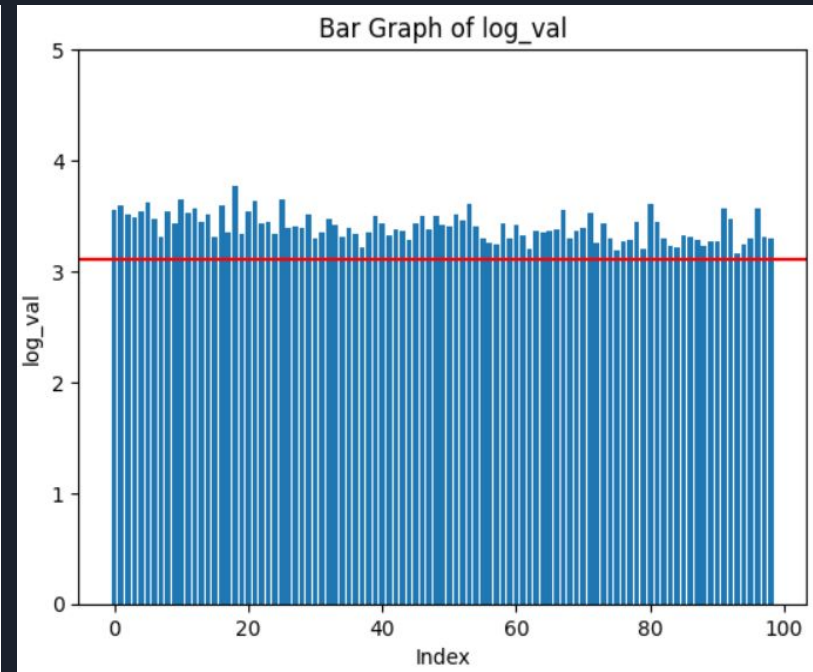
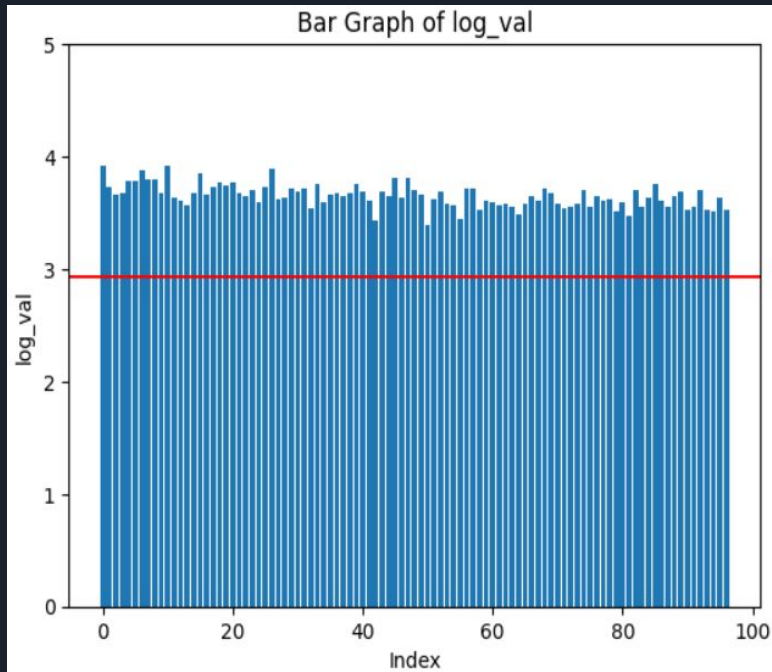
Output on Log-Likelihood using GPT2

	org_sentences	paraphrased	org_perplexity	para_perplexity	diff_perplex	log_likelihood_org	log_likelihood_para	diff_log_likelihood
0	Maryland's environmental protection agency is ...	Maryland's environmental protection agency is ...	15.579169	46.705853	31.126683	2.745935	3.843869	1.097935
1	Maryland's environmental protection agency is ...	Maryland's environmental protection agency is ...	15.579169	43.850727	28.271558	2.745935	3.780791	1.034857
2	Maryland's environmental protection agency is ...	Maryland's environmental protection agency is ...	15.579169	34.908882	19.329713	2.745935	3.552742	0.806807
3	Maryland's environmental protection agency is ...	Maryland's The state's environmental protectio...	15.579169	35.024487	19.445317	2.745935	3.556047	0.810113
4	Maryland's environmental protection agency is ...	Maryland's environmental protection agency reg...	15.579169	30.750065	15.170896	2.745935	3.425892	0.679957

Analysis on Log-Likelihood using GPT2



Continued.....





Future Work

- Other transformer models besides T5 and GPT-2, such as GPT-J, LLaMA, and XLNet, could be investigated for their efficacy in classifying sentences as human-written or AI-generated.
- Availability of platforms providing sufficient GPU resources is also crucial for carrying out future work smoothly.
- Additional research can explore the impact of different training datasets and fine-tuning strategies on the performance of transformer models for this task



Links to all the works

- [Link to Colab Notebook for text summarization](#)
- [Link to Colab Notebook for Text Classification](#)
- [Link to Colab Notebook for Paraphrasing](#)
- [Link to Colab Notebook for Perplexity Calculation](#)
- [Link to Colab Notebook for Log Likelihood](#)
- [Output Google Drive Link](#)



References

1. Mathur, S. (2021). Understanding T5 Model: Text to Text Transfer Transformer Model. Towards Data Science. Retrieved from <https://towardsdatascience.com/understanding-t5-model-text-to-text-transfer-transformer-model-69ce4c165023>.
2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. In Advances in neural information processing systems (pp. 5998-6008). <https://huggingface.co/t5-small>
3. Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... Brew, J. (2019). HuggingFace's Transformers: State-of-the-art Natural Language Processing. arXiv preprint [arXiv:1910.03771](https://arxiv.org/abs/1910.03771). <https://huggingface.co/docs/transformers/perplexity>
4. Chen, T., Wu, Y., Liu, S., Zhang, D., Zhao, T. (2020). A Transformer-based Approach for Distinguishing Human-written and Machine-generated Texts. arXiv preprint [arXiv:2010.11423](https://arxiv.org/abs/2010.11423)
5. Li, S., Guo, J., Zhang, Y., Liu, Y. (2021). AI Detection by Syntax: Can Syntax Help Distinguish between Human and Machine Text?. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) (pp. 4351-4360)
6. Schick, T., Schütze, H. (2021). Identifying and reducing gender bias in word-level language models. arXiv preprint [arXiv:2104.08786](https://arxiv.org/abs/2104.08786)
7. Yves Scherrer. 2020. TaPaCo: A corpus of sentential paraphrases for 73 languages. In Proceedings of the Twelfth Language Resources and Evaluation Conference, pages 6868–6873, Marseille, France. European Language Resources Association
8. Hugging face Dataset: TaPaCo



THANK YOU