Style text-to-transfer: Shakespearean Formal English to Modern English style transfer

CSCI 5800: Natural Language Processing and Generative AI – Fall 2024

Name	Student ID	Email Id
Anmol Singhal	111021672	Anmol.singhal@ucdenver.edu
Bhavishya Vudatha	111091008	bhavishya.vudatha@ucdenver.edu
Pooja Kulkarni	111024438	pooja.kulkarni@ucdenver.edu
Sai Krishna Karnam	110942676	saikrishna.karnam@ucdenver.edu

Problem Statement and Background:

Most readers today find it difficult to relate to classic literature, particularly Shakespeare's works. It is difficult to properly understand these masterpieces because of their old-sounding language, complicated sentence structures, and unfamiliar vocabulary. This is true, especially for students and educators in trying to fully engage with and understand these timeless writings. Finding a means to translate Shakespeare's writings into contemporary English while preserving their beauty and meaning will make them accessible and pleasurable for everyone.

The dataset used for this project contains paired sentences in Shakespearean English and their corresponding translations into modern English. Most of the data used for this project was sourced from litcharts, a public platform known for providing Shakespearean texts together with their modern English translations. Using Selenium, a web scraping tool, we collected paired sentences Shakespearean English and their corresponding modern interpretations. The scraping procedure was carefully planned to guarantee that we preserved the original language and its meaning while capturing highly accurate and quality data.

To further improve the dataset, we applied data augmentation techniques. One key method was used in generating more data is back translation, where modern English sentences were first translated into an intermediate language, such as French, and then back into English. This method successfully produced paraphrased copies of contemporary English sentences by introducing minor wording changes without sacrificing the essential meaning. By doing this, we expanded our dataset's size and diversity, which helped the model pick up a wider variety of linguistic patterns. Our model is trained and tested using this large, varied data set, which guarantees that it can handle a broad variety of language structures and styles.

To measure the success of our project, we will rely on well-known NLP evaluation metrics. During training, the model achieved a loss of 0.3453, which shows it's learning effectively. However, data by itself doesn't provide a complete picture. To ensure the translations are accurate and meaningful, we'll evaluate them using metrics like BLEU, which checks how closely our output matches the original meaning, and METEOR, which considers linguistic similarities. We'll also use BERTScore to compare the semantic meaning and CHRF, which focuses on character-level accuracy.

This problem is important to anyone interested in bridging the gap between historical texts and modern audiences. Students, teachers, literature enthusiasts and casual readers often struggle with classic literature. An easy translation preserves the charm of original works can greatly benefit these groups. This project has the potential to grow, with the possibility of making a system that can handle various kinds of formal texts and turn them into modern, easier-to-read languages. Our goal is to make classic literature more relatable and understandable for today's readers. We will also evaluate how accurate the conversions are using standard NLP metrics and may add more features if time allows.

The motivation comes from our own experiences as students, struggling to understand complex texts. This project has real-world value by showing how AI and NLP can help make difficult, old language easier to understand.

To conclude, the idea is to create a tool in which you put a sentence in Shakespearean English and our developed model will convert this classic text in modern English. The plan is to train the model to learn this behavior by analyzing the patterns of the style change and then convert text accordingly.

Methods

To convert Shakespearean English into Modern English, we explored several Natural Language Processing (NLP) techniques. Our main approach was to use Transformer-based models because they are highly advanced and effective for tasks like generating text and transferring writing styles.

File Structure and running: We began with two Excel files, test data and train data, located in the data preprocessing folder, which were cleaned through preprocessing steps in Data_Transform.ipynb, resulting in cleaned_test.xlsx and cleaned_train.xlsx. These cleaned files were then merged into a single dataset, new_data, stored in the data augmentation folder, and underwent data augmentation using NLP_Style_Transfer.ipynb, producing translations.xlsx as the final consolidated dataset. This dataset was used for fine-tuning and training the model, processed using Copy_of_FinalCode.ipynb, with the trained model saved in the Frontend folder. The Frontend folder contains the full deployment setup, including the model (in the model folder), the database for user information and history (in the instance folder), UI templates (in the templates folder), style.css for styling (in the static folder), and the Flask application code (app.py) for integrating the model and running the website on a local server. The website enables seamless translation of Shakespearean English into Modern English.

```
├── Data augmentation
│   ├── new_data.xlsx
│   ├── NLP_Style_Transfer.ipynb
│   └── Translation.xlsx
├── Data pre processing
│   ├── Cleaned_Test_Data.xlsx
│   ├── Cleaned_Train_Data.xlsx
│   ├── Data_Transform.ipynb
│   ├── test.xlsx
│   └── train.xlsx
├── Fine tuning and training
│   ├── Copy_of_FinalCode.ipynb
│   └── Translation.xlsx
└── FrontEnd
    ├── app.py
    ├── instance
    │   └── database.db
    ├── model
    │   ├── added_tokens.json
    │   ├── config.json
    │   ├── generation_config.json
    │   ├── model.safetensors
    │   ├── special_tokens_map.json
    │   ├── spiece.model
    │   └── tokenizer_config.json
    ├── static
    │   └── style.css
    └── templates
        ├── About.html
        ├── Base.html
        ├── History.html
        ├── Index.html
        ├── Login.html
        └── Register.html
```

Data Preprocessing:

Data Overview (Before Preprocessing): Total Test Data: 1,002 rows , Total Train Data: 69,236 rows

1. **Data Cleaning and Handling Missing Values:**

- **Punctuation Handling**: We identified all special characters (**Test Data:** {'é', '”', '!', ':', '‘', ';', '.', 'è', ']', '(', '"', '-', '’', '—', '"', ',', 'Ô', '?', ')', '[', '“'} **Train Data:** {'é', 'ï', '–', '”', '!', '&', ':', 'ç', '…', '‘', ';', '.', 'æ', 'à', 'è', ']', '(', 'â', '"', 'î', '-', '’', '—', ',', '"', 'Ô', '?', ')', 'ê', '$' }) in the dataset and prepared a replacement dictionary to normalize them. Special characters such as accented letters (e.g., é, â) were converted to their nearest ASCII equivalents (e.g., é to 'e'). However, punctuation marks that convey important information, like commas, parentheses, or question marks, were retained, as they contribute to the dialect and meaning of the sentences.
- **Lowercasing**: All text is converted to lowercase except for words that are fully capitalized. This normalization step helps the model to treat words like "King" and "king" to simplify the vocabulary.
- **Handling Missing Data**: Any rows that have missing translations in either the Shakespearean or Modern English column are removed to prevent discrepancies during training.
- **Duplicate Removal:** We found 698 duplicates in training data and 2 rows in test data. We removed these duplicates from both training and test data. We also checked how many duplicates were there in both test and train data and removed 25 duplicates from the test dataset.

2. **Addressing Space and Joint Words**:
   - **Detecting and Correcting Missing Spaces**: We identified patterns where two words were incorrectly joined without a space (e.g., "unfoldHis"). We corrected these issues using regex-based pattern matching, to ensure each word is properly separated, maintaining semantic clarity. Shakespearean data had about 3 records, and Modern English data had about 2 records with such issues.
   - **Whitespace Removal**: We removed all the extra spaces before and after word in both the train and test dataset.

3. **Text Normalization**:
   - **Unicode Normalization**: Special characters such as accented letters (for e.g., é, â) were normalized using a combination of dictionary replacement and Unicode normalization techniques. First, we identified all special characters in the dataset and used a dictionary to replace them with their nearest ASCII equivalents (e.g., é to "e"). Later, we applied Unicode normalization to ensure any remaining accented characters were handled appropriately. A quick test confirmed that no non-ASCII characters were missed during this process.
   - **Expanding Contractions**: Both Shakespearean and Modern English contain contractions, we expanded them for example, "you've" to "you have" using the contractions library. This was done only for Modern English, ensuring that the stylistic elements of Shakespearean English are preserved.

4. **Data Splitting**: Once pre-processing is complete, the data will be split into training, validation, and test sets to ensure that the model learns effectively and generalizes well.

Data Overview (After Preprocessing): Total Test Data: 975 rows, Total Train Data: 68,538 rows

Data Augmentation:

To make the dataset more robust, we used back translation as a data augmentation strategy. Here’s how it worked:

- **Back Translation Using MarianMT Model Transformer**: We randomly selected 20,000 records from the dataset containing Shakespearean English and Modern English translations. To paraphrase the Modern English sentences, we used the MarianMT Model Transformer model for back translation. During the process, Modern English was translated into French, an intermediate language, and finally back to English. This method successfully paraphrases the text by generating sentences with semantically identical content but different wording and structure.

- **Post-Translation Preprocessing**: After back translation, we cleaned the augmented data to ensure that it was consistent and did not include any duplicates. Such activities included checking for identical sentences between the back-translated and the original Modern English texts; making sure that there were no NULL records in the dataset.

By performing this augmentation, we increased the dataset's size and diversity, providing the model with more examples.

Pre-trained Transformer Models:

The core of our approach was fine-tuning the T5 (Text-to-Text Transformer) model, T5 is a highly flexible model that treats every task as a text-to-text problem. This made it a perfect fit for translating Shakespearean English into Modern English. The model does not just transform words, it preserves the meaning and style of the text, which is crucial for our goal.

By training the model on our curated and augmented dataset, we were able to teach it the patterns of Shakespearean language and the appropriate Modern English equivalents.
We picked T5 because it is a powerful transformer model trained specifically to perform tasks such as language translation, summarization, and other text-to-text related tasks. Specific to Translation, to finetune and test the model, "translate Early English to Modern English: " was used to specify the task to the T5-small model for every sentence. The T5 model (extended over BERT) performs better than other models due to its single integrated model trained on a massive dataset (C4-Colossal Clean Crawled Corpus) and achieved state-of-the-art results on many benchmarks. The model randomly chooses words for corruption in a sentence and the output sequence consists of the dropped-out spans delimited by the tokens that replaced the corrupted words.

Process:

1. We took the data prepared from web scraping (and later data augmentation) and converted it to a DataFrame, then split it into testing and evaluation datasets (90-10).
2. The input sentences are appended with the "translate Early English to Modern English: " string to specify the task to the model.
3. Then the input and target sentences are truncated / padded to a constant size and tokenized using the T5 tokenizer.
4. Trainer API was used to finetune the model and specify the hyperparameters in TrainingArguments.
5. After the training is completed, a test dataset not matching with the training and evaluation dataset has been used to extract translated sentences using the finetuned model and compute the evaluation metrics.

Parameters: The following hyperparameters were used throughout the training process:

- **Number of Epochs**: The model was trained for 5 epochs, which provided a balance between learning from the data and avoiding overfitting.
- **Learning Rate**: Set to 5e-5 to make sure for stable convergence during training without overshooting the optimization minima.
- **Batch Size**:
  Training: 16 examples per device.
  Evaluation: 16 examples per device.
- **Weight Decay**: A regularization parameter of 0.01 was used to minimize overfitting by penalizing large weights.
- **Evaluation Strategy**: The evaluation of the model was conducted at the end of each epoch to monitor validation performance.

Evaluation Metrics:
We have assessed the performance of the model for Formal English to Modern English translation using multiple metrics such as BLEU Score, METEOR Score, BERTScore and ChrF(Character n-gram F-score). Each metric evaluates various aspects of the translation quality to ensure a comprehensive analysis of the model's performance. Below is the description of the metrics used:

1. **ChrF:** The ChrF (Character n-gram F-score) measures translation quality by evaluating character n-grams rather than word n-grams. It is particularly suited for translations involving morphological and structural differences. The sacrebleu.metrics.CHRF implementation was used to compute ChrF scores for each sentence. The final reported score is the average ChrF score across all sentences in the dataset.

2. **BERTScore: The** BERTScore evaluates the semantic similarity between the predicted and reference sentences using contextual embeddings generated by pre-trained BERT models. This metric captures semantic nuances beyond simple lexical overlap. The bert_score function from the BERTScore library was used to compute Precision, Recall, and F1 scores for each sentence pair. The mean F1 score across all sentences is reported as the final BERTScore.

3. **BLEU Score:** The BLEU (Bilingual Evalution Understudy) score measures the overlap between the predicted translation and reference text at the n-gram level. A higher BLEU score indicates closer resemblance to the reference text. Here, we have implemented BLUE score where we are able to calculate each sentence using NLTK's sentence_bleu function with smoothing technique (method4) to handle edge cases with zero probabilities for certain n-grams. The final score of individual sentence level BLEU score across the dataset.

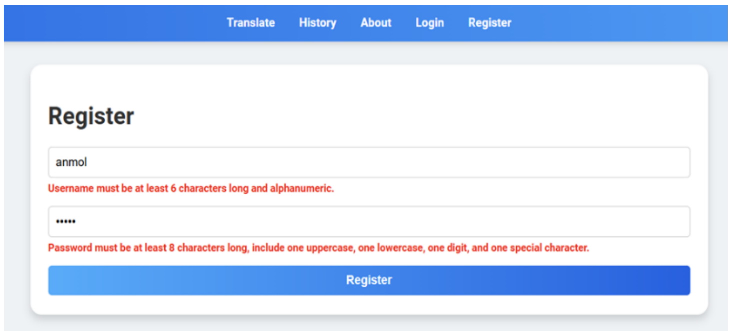
4. **METEOR Score:** The METEOR (Metric for Evaluation of Translation with Explicit Ordering) score evaluates translations by considering matches based on exact, stemmed, synonym, and paraphrase matches, alongside a

penalty for word order violations. METEOR scores were calculated for each sentence using NLTK's single_meteor_score function. The final reported score is the average METEOR score across all sentences in the dataset.

UI: We have built an interactive UI in order to let users translate the sentence easily. We have 5 pages with login and logout option on the website. Translate Page: Let users input Shakespearean English text and receive real-time Modern English translations. It includes a text box for input, and a Translate button, and a section for displaying both the original and translated text. History Page: Let users view a list of all their past translations. About Page: provides information about the project, its ideology, and the team's contributions.

Login Page: registered users can log in with credentials. Validation rules make sure that the username and password fields are not left blank, and error messages are displayed for invalid inputs. Register Page: Gives new users a place to create accounts, Validation rules: usernames must be at least six characters long and alphanumeric, passwords must be at least 8 characters long, including a number, an uppercase letter, a lowercase letter, and a special character.

The frontend design is structured with HTML templates located in the templates folder and styled using style.css from the `static` folder for a clean and consistent appearance. The Flask backend `app.py` connects the UI to the trained model and database and rest of the website. The trained model processes the translations, while the database.db file stores user details in the User table and translation history in the Translation History table. The application is deployed on a local server.



Register

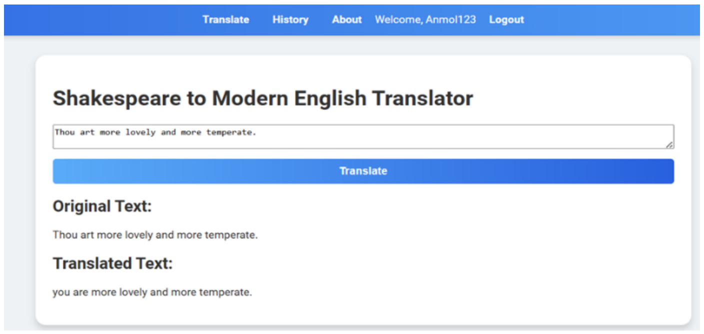
anmol

Username must be at least 6 characters long and alphanumeric.

•••••

Password must be at least 8 characters long, include one uppercase, one lowercase, one digit, and one special character.

Register



Shakespeare to Modern English Translator

Thou art more lovely and more temperate.

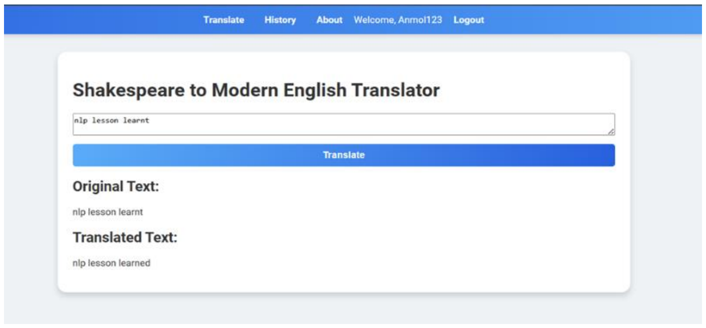
Translate

Original Text:

Thou art more lovely and more temperate.

Translated Text:

you are more lovely and more temperate.



Shakespeare to Modern English Translator

nlp lesson learnt

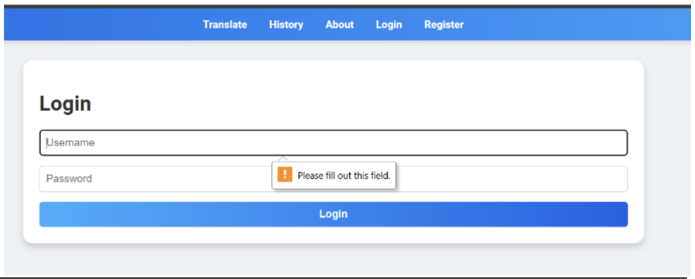
Translate

Original Text:

nlp lesson learnt

Translated Text:

nlp lesson learned



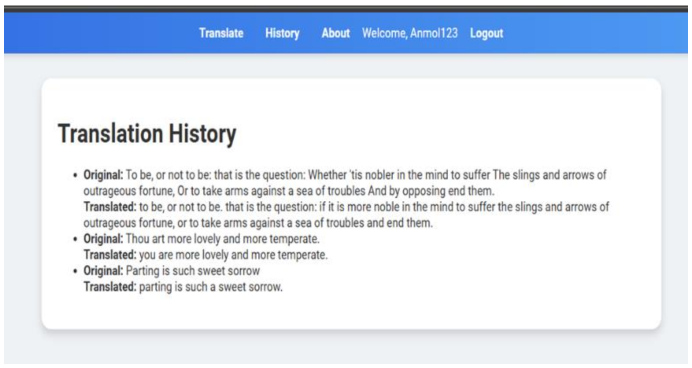
Login

Username

Password

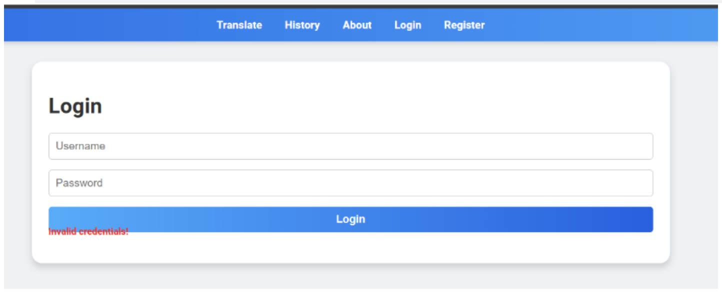
Please fill out this field.

Login



Translation History

- **Original:** To be, or not to be: that is the question: Whether 'tis nobler in the mind to suffer The slings and arrows of outrageous fortune, Or to take arms against a sea of troubles And by opposing end them.
  Translated: to be, or not to be. that is the question: if it is more noble in the mind to suffer the slings and arrows of outrageous fortune, or to take arms against a sea of troubles and end them.
- **Original:** Thou art more lovely and more temperate.
  Translated: you are more lovely and more temperate.
- **Original:** Parting is such sweet sorrow
  Translated: parting is such a sweet sorrow.



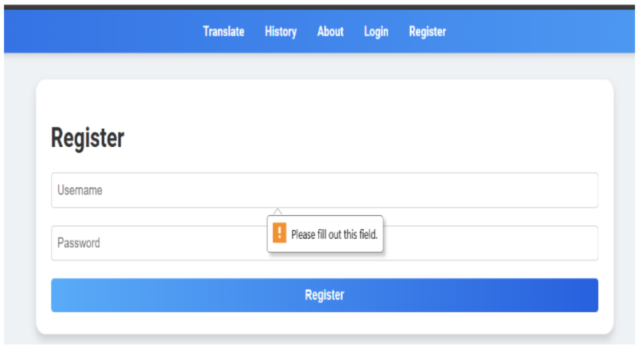
Login

Username

Password

Login

Invalid credentials!



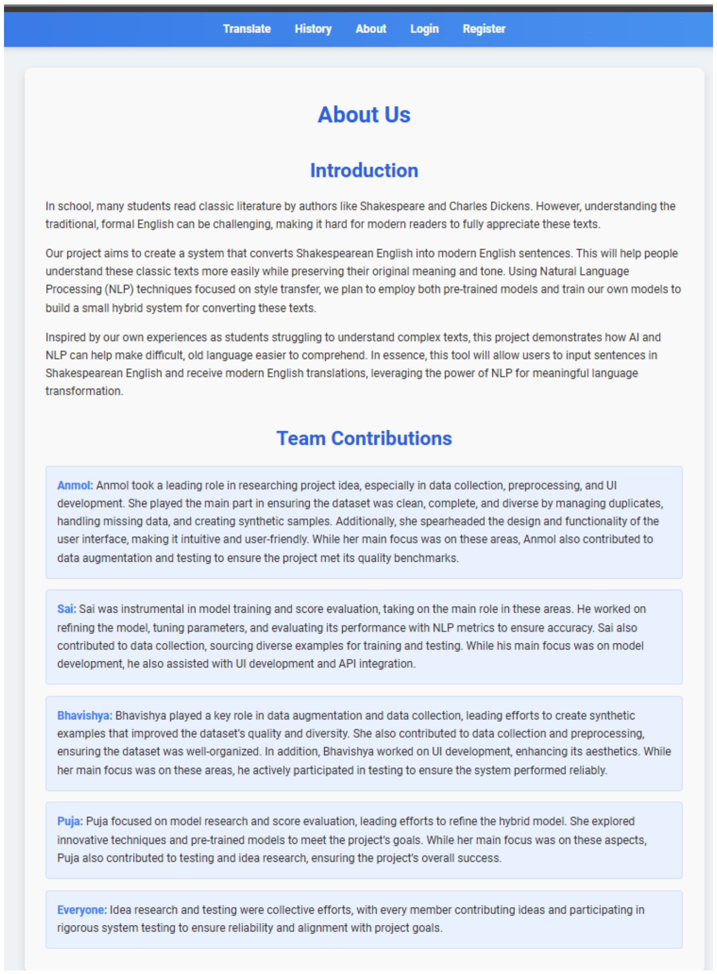
Register

Username

Password

Please fill out this field.

Register



About Us

Introduction

In school, many students read classic literature by authors like Shakespeare and Charles Dickens. However, understanding the traditional, formal English can be challenging, making it hard for modern readers to fully appreciate these texts.

Our project aims to create a system that converts Shakespearean English into modern English sentences. This will help people understand these classic texts more easily while preserving their original meaning and tone. Using Natural Language Processing (NLP) techniques focused on style transfer, we plan to employ both pre-trained models and train our own models to build a small hybrid system for converting these texts.

Inspired by our own experiences as students struggling to understand complex texts, this project demonstrates how AI and NLP can help make difficult, old language easier to comprehend. In essence, this tool will allow users to input sentences in Shakespearean English and receive modern English translations, leveraging the power of NLP for meaningful language transformation.

Team Contributions

Anmol: Anmol took a leading role in researching project idea, especially in data collection, preprocessing, and UI development. She played the main part in ensuring the dataset was clean, complete, and diverse by managing duplicates, handling missing data, and creating synthetic samples. Additionally, she spearheaded the design and functionality of the user interface, making it intuitive and user-friendly. While her main focus was on these areas, Anmol also contributed to data augmentation and testing to ensure the project met its quality benchmarks.

Sai: Sai was instrumental in model training and score evaluation, taking on the main role in these areas. He worked on refining the model, tuning parameters, and evaluating its performance with NLP metrics to ensure accuracy. Sai also contributed to data collection, sourcing diverse examples for training and testing. While his main focus was on model development, he also assisted with UI development and API integration.

Bhavishya: Bhavishya played a key role in data augmentation and data collection, leading efforts to create synthetic examples that improved the dataset's quality and diversity. She also contributed to data collection and preprocessing, ensuring the dataset was well-organized. In addition, Bhavishya worked on UI development, enhancing its aesthetics. While her main focus was on these areas, he actively participated in testing to ensure the system performed reliably.

Puja: Puja focused on model research and score evaluation, leading efforts to refine the hybrid model. She explored innovative techniques and pre-trained models to meet the project's goals. While her main focus was on these aspects, Puja also contributed to testing and idea research, ensuring the project's overall success.

Everyone: Idea research and testing were collective efforts, with every member contributing ideas and participating in rigorous system testing to ensure reliability and alignment with project goals.

One fascinating observation we made while testing our model was its ability to adapt beyond Shakespearean English. For instance, in one of the screenshots, we input the phrase "NLP lessons learnt," which is not Shakespearean English. However, the model still converted "learnt" to "learned," demonstrating its ability to handle single-word transformations effectively. This highlights the model's potential to assist with other English language styles, providing a broader range of applications beyond just Shakespearean-to-Modern English translation.

Baseline Approaches:
We compared the performance of our fine-tuned T5 model to a simple baseline method. We used a rule-based system that replaced Shakespearean words with Modern English equals from a predefined dictionary (e.g., "thou" became "you,"). This approach was able to handle simple word replacements, but it struggled with more complicated sentence structures, and overall meaning of the text.

Results And Observations:

Training and Validation Loss:

The model was fine-tuned using the T5 transformer architecture with the following hyperparameters:

- Learning Rate: 5e-5, Weight decay: 0.01
- Batch size per device: 16 (for both training and evaluation)
- Number of epochs: 5

Results per epoch (Before Data Augmentation):

Epoch	Training Loss	Validation Loss
1	0.4245	0.385983
2	0.4123	0.375738
3	0.396	0.370065
4	0.3915	0.367971
5	0.3942	0.367195

Results per epoch (After Data Augmentation):

Epoch	Training Loss	Validation Loss
1	0.3680	0.3377
2	0.3510	0.3274
3	0.3453	0.3222
4	0.3357	0.3196
5	0.3347	0.3190

Above two tables and figures 1 and 2, show the training and validation loss over 5 epochs. Loss converges towards the end in both scenarios. The declining validation loss over epochs indicates effective generalization and minimal overfitting.

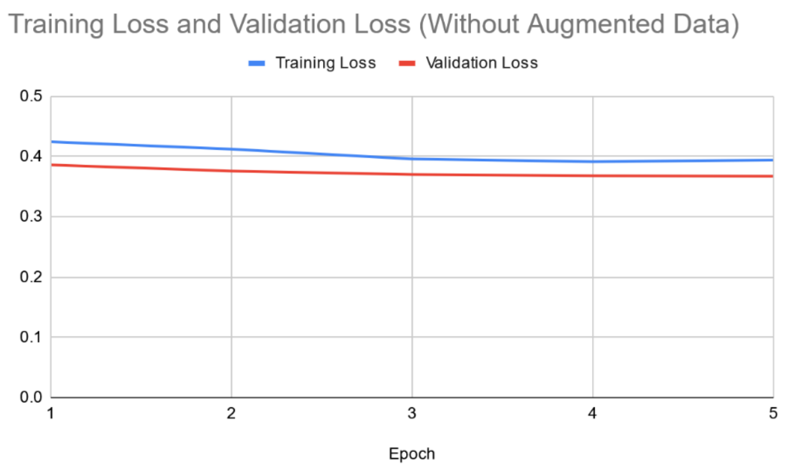


Figure 1: Plot of Epochs vs Model Loss (Train and Evaluation) without adding Augmented Data

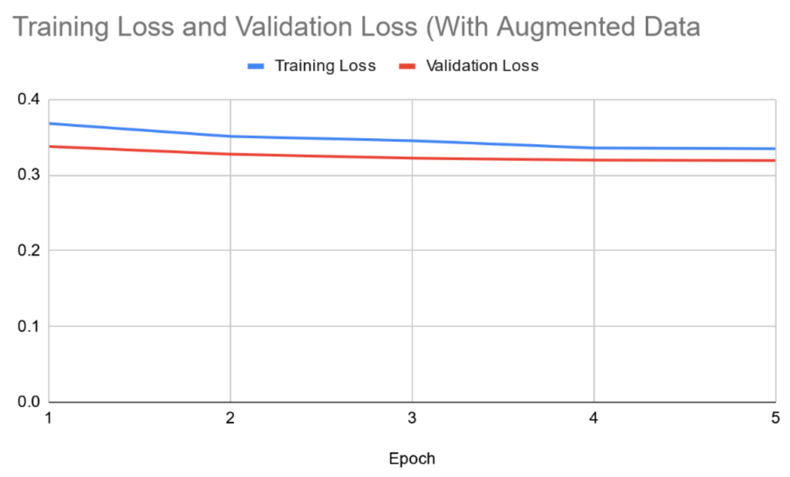


Figure 2: Plot of Epochs vs Model Loss (Train and Evaluation) with added Augmented Data

Below are the scores achieved from our model.

Metric	**Evaluation Score**		**Insights**		**Feedback**
	Before Augmentation	**After Augmentation**	**Before Augmentation**	**After Augmentation**	
ChrF Score	27.01	36.36	A low ChrF score was observed, meaning the character-level accuracy could be improved.	The score significantly increased, suggesting data augmentation enhanced character-level translation.	A good ChrF score shows that the model captures character-level accuracy effectively, especially for morphologically complex translations
BERTScore (F1)	0.88	0.9015	The BERTScore was slightly lower, indicating some semantic gaps between predicted and reference texts.	The score improved, showing better alignment in terms of contextual semantic meaning.	A high BERTScore reflects the model's strong semantic understanding and contextual alignment with reference texts. This is a very good result
BLEU Score	0.0478	0.0952	The BLEU score was very low, indicating a poor overlap between predicted and reference n-grams.	The score improved significantly, suggesting that data augmentation helped increase lexical matching.	The BLEU score is low, suggesting the model struggles with strict n-gram matching. This metric may not fully capture semantic accuracy for this task.
METEOR Score	0.2086	0.3093	A relatively low METEOR score indicated moderate alignment with the reference texts.	The score increased, suggesting better paraphrase and synonym handling with augmented data.	A moderate METEOR score indicates some alignment with reference texts, considering synonyms and paraphrasing. Shows room for improvement in alignment

Challenges and Methods That Did not Work:

- **Back Translate with Google API:** During the data augmentation process, we tried using googletrans and deeptranslate for back translation. These tools initially appeared to be a solid choice, but we soon faced difficulties. The APIs frequently timed out, translations were inconsistent, and rate limits slowed us down significantly. These issues made it clear that these tools weren't good choices for handling a large dataset like ours, so we had to explore other options.
- **Augmentation with Synonym Replacement**: Another data augmentation technique was replacing words with synonyms using WordNet. While this increased data diversity, the semantics of the sentence were not captured.

Tools

Our primary platform will be Google Collab as it has the most accessible and powerful computational resources like GPU support. Using this we have planned to **develop our own model**, and use models like sequence-2-sequence, and fine-tune existing mT5 model or some other models to translate classic Shakespearean English text to Modern English text. We have planned to use several libraries for data processing, model training and evaluation like transformers, Py Torch and many more. We will be using Kaggle and web scrapping to develop our dataset collection. We will perform multiple preprocessing steps to prepare the data for training and ensure consistency. This will include tasks such as cleaning the text by removing unnecessary punctuation, converting it to lowercase, eliminating extra spaces, and filtering out non-alphanumeric characters. We will apply word and subword tokenization techniques (like Byte-Pair Encoding) to break the text into manageable units for transformation models. Additionally, spelling normalization will be conducted to replace archaic words with their modern counterparts (like "thou" to "you"). We have used python, flask, html, CSS in order to develop the front end and sqlite.

Lessons Learned:

The project of translating Shakespearean English into Modern English was a difficult but rewarding one. It taught us much about the complexity of language and how powerful NLP techniques are today; it also taught us about careful data handling and realistic expectations.

One of the biggest lessons we learned was just how important good data preprocessing is. Shakespeare's language, with its unique punctuation, archaic spellings, and stylistic quirks, presented numerous challenges. For example, we had to carefully handle missing spaces between words, normalize special characters, and even expand contractions in modern English—all to make the data as clean and consistent as possible. This hard work paid off, as it allowed the model to learn and generalize better.

Fine-tuning T-5 was another major milestone. It was exciting to see the model gradually learn how to translate complex Shakespearean sentences into fluent, modern English. However, we discovered that the quality of the dataset was just as important as the model itself. A well-balanced data set with diverse examples made a significant difference in the model's ability to handle a wide variety of inputs.

The use of data augmentation improved model performance significantly. By adding paraphrased versions of Modern English sentences through back translation, we expanded the training data without having to collect more samples. This gave the model exposure to different sentence structures and phrasing styles, which made it more robust and adaptable. After adding data augmentation to the pipeline, we saw clear improvements in the model's evaluation scores.

In the end, this project taught us how crucial it is to have clean, diverse data and to leverage techniques like data augmentation. It also showed us the potential of advanced models like T5 when paired with thoughtful preparation. These lessons will definitely shape how we approach future NLP projects.

Team Contributions:

Name	**Work division**
Anmol Singhal	Data preprocessing, Data Augmentation, UI, API integration, Idea research
Bhavishya Vudatha	Data Augmentation, Data preprocessing, UI, API integration, Idea research
Pooja Kulkarni	Score evaluation, Model Research, Idea research, Model,
Sai Krishna Karnam	Model, Score evaluation, UI, API integration, Idea research

References:

1. litcharts, Shakescleare - Every Shakespeare play, poem, and sonnet alongside a modern English translation everyone can understand.
2. https://medium.com/@ageitgey/build-your-own-google-translate-quality-machine-translation-system-d7dc274bd476; Build Your Own 'Google Translate'-Quality Machine Translation System
3. https://arxiv.org/pdf/2010.11934; mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer
4. https://huggingface.co/docs/transformers/v4.17.0/en/tasks/translation, Translation
5. Roudranil/finetuning-llms-for-conversation-in-shakespearean-english (github.com)
6. Roudranil/shakespearean-and-modern-english-conversational-dataset · Datasets at Hugging Face