Assessing the Efficacy of Synthetic Data for Enhancing Machine Translation Models in Low Resource Domains

MSc Research Project
Data Analytics

Shweta Yadav
Student ID: x21209251@student.ncirl.ie

School of Computing
National College of Ireland

Supervisor: Prashanth Nayak

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Shweta Yadav
Student ID:	x21209251@student.ncirl.ie
Programme:	Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Prashanth Nayak
Submission Due Date:	12/08/2023
Project Title:	Assessing the Efficacy of Synthetic Data for Enhancing Machine Translation Models in Low Resource Domains
Word Count:	4626
Page Count:	15

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Shweta Yadav
Date:	18th September 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Contents

List of Figures

List of Tables

Assessing the Efficacy of Synthetic Data for Enhancing Machine Translation Models in Low Resource Domains

Shweta Yadav
x21209251@student.ncirl.ie

Abstract

An artificially generated dataset mimics real-world data in terms of its statistical properties, but it contains no real information. Data around rare occurrences like Covid-19 pandemic is difficult to capture in real-world data due to their infrequent nature. Additionally, cost involved and time-consumption to gather real world data is a big challenge. In such cases, synthetic data can help create more balanced datasets for model training. This project investigates the effectiveness of using synthetic data for tuning machine translation models when training data is limited. The Covid-19 domain is chosen considering the urgency and importance of the global accessibility of information related to the pandemic. TICO-19, a publically available dataset was effectively formulated to cater to this need. The medical terminologies were extracted and passed to OpenAI API to generate training language pair data. The fine-tuned Davinci model is then verified with blind test data provided under TICO-19 for translation from English to French. SacreBLEU score is used to compute the translation quality, the fine-tuned model has a significantly higher BLEU score of 19.54 in comparison to the base model with a BLEU score of 0.44. The adapted model also has a comparable score to the next-generation version of davinci with a BLEU score of 22.29.

Keywords – OpenAI, davinci, TICO-19, low resource domain, machine translation, Covid-19

1 Introduction

The introduction of the attention layer in transformer-based models (Vaswani et al. 2017) transformed the field of Natural Language Processing (NLP), there is a drastic paradigm shift when choosing models for text processing from RNNs and CNNs. Further, the study by Bahdanau et al. (2014) introduced Neural Machine Translation (NMT) which showed improvement over traditional phrase-based translation. Machine Translation (MT) is the branch of NLP problems where a sentence from one language is translated into another using a computer application [1], while carefully considering the rules of both source and target language. NMTs have worked considerably well when a resource-rich language pair is chosen, however, it still faces the challenges associated with low resource domain

[1] `https://phrase.com/blog/posts/machine-translation/`

(Koehn & Knowles 2017). A language is treated as low-resourced when it lacks linguistic resources or monolingual/bilingual corpora required for training models.

Collating adequate corpus for ML models is already a challenging task, however, when using transformers, which need even large data for training, working with low-resource language becomes even more difficult. Adapting the NMT models, which are already trained on heavy resources, to domain-specific translation has been extensively researched to overcome the issue of poor in-domain translation of these existing models. Research by Kumar et al. (2021) attempts to provide a framework where a model already trained for one language pair can be extended to another. The study by Luong & Manning (2015) presented the improvement that can be seen in fine-tuning a previously trained model with a domain-specific corpus. However, adapting models with domain-specific translations showed promising results, but the issue associated with the lack of training data still remains.

With the introduction of generative AI and successful models like OpenAI GPTs [2] have made generating synthetic data more accurate. While the GPT models can quickly generate *completion* based on the *prompt* they are given, the model loses the accuracy of the text it generates when generating longer sentences and becomes more random. The study by Moslem et al. (2022) discusses the efficiency and effectiveness of adapting pre-trained language models to a domain-specific MT. It presents a text-generation technique that produces domain-specific sentences and extensively verifies the feasibility of synthetic data for training MT for domains with low to no parallel datasets.

In this paper, a motivated approach to utilizing domain-specific data augmentation is introduced for training language models for machine translation. The study leverages the GPT-3.5 model of OpenAI for generating the synthetic data and verifies how well an existing model can be enhanced using this near-real data. The use of synthetic data also caters to the ethical issues that arise with using actual data in sensitive domains like healthcare and the challenges of working with domains with low monolingual or bilingual data. Generative models need a term or sentence to generate a completion for them; in this study, terminologies related to COVID-19 are extracted from the TICO-19 dataset, which was collated to advance and promote the study of machine translation of pandemic-related information.

The rest of the paper is organized as follows. In Section 2, a detailed discussion of the related work is done. Then, the methodology is presented in Section 3. Section 4 details the configuration and system design. In Section 5, a detailed view of conducting the technical aspect of research is given. Section 6 and Section 7, describe the experiments performed to verify the model's translation quality and the results of the experiments, respectively. Finally, the conclusion of the paper and discussion of future work are done in Section 8.

2 Related Work

The need for machine translation has been ever present since the advent of the internet. With an efficient Machine Translation (MT) system, the circulation of important information will become more effective and feasible. Apart from being a medium of making a piece of information globally accessible, Yu et al. (2018) presented translation as an effective method of data augmentation. Techniques like back translation can help generate

[2] `https://platform.openai.com/docs/introduction/key-concepts`

a corpus for low-resource language from a resource-rich monolingual corpus. However, the study showed significant results for the English and French language pair which is a resource-rich language pair but when a motivated approach (Amjad et al. 2020) was tried for English to Urdu translation for training a model for fake news detection in Urdu, the results were poor. This shows the efficiency of MT models largely depends on the language pair selected for research and low-resource language pairs will always be at a loss. On a high-level MT methodology can be divided into two, rule-based approach and corpus-based approach. A rule-based approach requires a huge set human annotated rules (distinct for every language pair) that will be fed to the system for it to learn while in a corpus-based approach, the machine learns the features from the parallel corpus (source-target language pair) (Okpor 2014). Two of the further classification of Corpus-Based Machine Translation (CBMT) are Statistical Machine Translation (SMT) and Neural Machine Translation (NMT).

In contrast to SMT (Koehn et al. 2007), NMT (Bahdanau et al. 2014, Cho et al. 2014, Sutskever et al. 2014) allows for end-to-end training of a translation system without requiring complicated decoding algorithms, word alignments or translation rules. MT for low-resource domains has poor performance due to a lack of parallel corpora, however, the need for domain-specific translation is significantly more than the generalized translation models. Domain adaptation of SMT models has been researched in lengths (Koehn & Schroeder 2007, Bertoldi & Federico 2009) but due to their dissimilar characteristics, these methods cannot be applied to NMT models.

Domain adaptation of NMT is being thoroughly researched and due to the availability of a wide array of language models (LMs), it has become fairly feasible. The use of pre-trained models due to their capacity to identify a wide range of linguistic features without having to train them from scratch, which can be computationally challenging and data-intensive, but can yield better results has significantly increased. Such models are already trained on basic textual features and are available for use-case-specific fine-tuning. The architecture considered for an NMT model consists of three layers, an encoder, a decoder, and an attention layer, due to these complex and data-intensive layers training an NMT requires a large amount of parallel corpus and lack of this degrades the performance and could result in overfitting of the models. One of the methods for data-centric domain adaptation of NMT is fine-tuning a generalized MT with in-domain corpus, however, this method has its drawbacks. The study by (Dakwale & Monz 2017) discusses this overfitting and proposes a solution to prevent the degradation of out-of-domain translations once a model is fine-tuned. Another way of fine-tuning a model is through synthetic parallel corpora. A study by Sennrich et al. (2015*a*) discusses the use of back translation on monolingual datasets to generate parallel corpus and tune the NMT model. There have been multiple types of research done discussing the effectiveness of generating synthetic data for domain adaptation with either source-side monolingual data (Sennrich et al. 2015*b*), target-side monolingual data (Zhang & Zong 2016) or both (Park et al. 2017). These studies show how effective synthetic data can be for fine-tuning a model.

A study by Carvajal-Patiño & Ramos-Pollán (2022) discusses the benefit of using synthetic data for enhancing predictions. The methodology fed the real training data to generative models to generate synthetic data and verified the percent combination of real and synthetic data when evaluating the model predictions. The results were positive however the simplicity of the generative model used resulted in imbalanced data which could be further improved by using more advanced generative models available. The benefits, challenges and risks of utilizing synthetic data are discussed by James et al. (2021). However, there will always be resentment towards accepting such data specifically in the healthcare domain, the study explores the benefits and presents a cautious way of gaining acceptance for this data. The evaluation metrics presented by Yale et al. (2020), developed for evaluating synthetic data in the healthcare domain, verify the data in terms of resemblance with original data, privacy, utility, and footprint. Data generated can be evaluated by utilizing this metric and the generation method can be verified. ChatGPT has taken over the field of generative AI like a storm. Being trained on billion data points its ability to produce human-like responses has been its biggest selling point. The study by Javaid et al. (2023) explores the benefits of leveraging ChatGPT in healthcare domain while also discussing its drawbacks and ethical issues pertaining to its usage. It highlights how well the data generated can be used for various NLP tasks while also emphasizing the risks of false information. Synthetic data especially in a domain as sensitive as healthcare will always raise privacy and ethical concerns but it can also offer several potential benefits when approached thoughtfully and with appropriate safeguards.

3 Methodology

In the research, a study-specific version of KDD is followed, the phases of KDD are refined and utilized to make it more accurate with the steps undertaken while conducting the study. The steps followed are data gathering, data pre-processing, data transformation, model fine-tuning, and evaluation. The steps are depicted in the Figure 1 below:

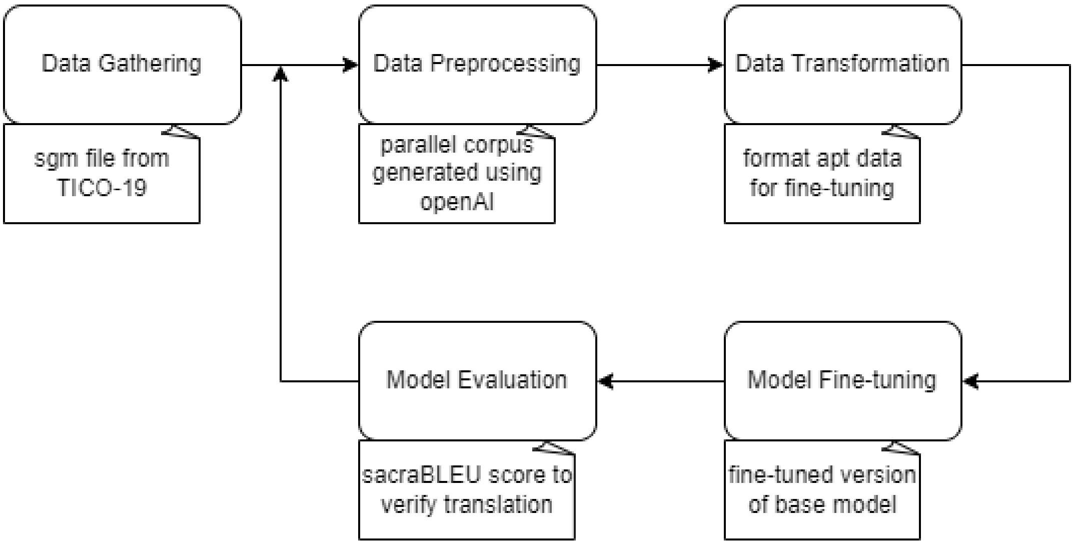
Data Gathering
sgm file from TICO-19
Data Preprocessing
parallel corpus generated using openAI
Data Transformation
format apt data for fine-tuning
Model Evaluation
sacraBLEU score to verify translation
Model Fine-tuning
fine-tuned version of base model

Figure 1: Research Methodology

1. Data Gathering: The main idea behind the research is to verify if advancements in low-resource domains can be supported by synthetic data. TICO-19 dataset is a resource specifically collated to further the research in the field of machine translation. The dataset contains sgm files containing Covid-19 specific terms in both source and target languages. The dataset also contains a test set containing language pairs for 9 rich resource languages and 26 low resource languages (Anastasopoulos et al. 2020) which will act as the ground truth and be compared while

evaluating the models. The pair considered for this research is English to French. The terminologies are extracted and stored for further processing.

2. Data Preprocessing: The next step is to generate the parallel dataset, the terms extracted are sent to openAI API to generate sentences for these terms. There are only 215 unique terms in the dataset, in order to increase the volume of training data, five statements are generated for each term. While generating the sentences there are various parameters that are tuned to have more variant and contextually sound datasets. These statements are then sent to API in order to be translated into the target language. With the completion of this step, the training data is generated, having domain-specific terms, synthetic source language sentences, and translated target language sentences, also called parallel corpus for machine translation. **davinci** is a powerful and sophisticated language model developed by OpenAI. It is part of the GPT-3 (Generative Pre-trained Transformer 3) family of models and has been trained over 175 billion parameters. This API for using and fine-tuning the davinci model is publicly available for research. The API call includes the below-mentioned parameters that help in controlling and modifying the type of data generated:

   (a) stop: This parameter defines the end of the output generated. davinci's output does not include any end character for its completion key, hence, the parameter is set to None but this should be updated to the character added at the end of the completion key of fine-tuning data prompted when generating jsonL data.

   (b) temperature: This parameter defines the randomness of the text generated by the API call. The value ranges from 0.0 to 1.0, a high value will result in more diverse and creative responses, but they might be less coherent or accurate. On the other hand, a lower temperature value, such as 0.2, will produce more focused and deterministic but duplicate responses.

   (c) prompt: This parameter defines the input that is passed to the API, for which the model should generate the completion. It is the combination of the action to be performed followed by the text on which the action is to be performed, in case no action is provided, the call will add random keywords following the input text/term.

   (d) max_tokens: This parameter limits the number of keywords or tokens that are to be generated by the model. This is how one can limit the maximum length of the output. There is no parameter that could decide the minimum length of the output. For generative AI, the low value of the token being generated results in better-formed sentences in comparison to when instructed to form a longer sentence.

   (e) engine: This parameter define the model that should perform the required task. Few of the OpenAI model's available are *davinci, curie, babbage, text-davinci-002, text-davinci-003 etc* [3]

3. Data Transformation: Once the parallel corpus is in place, the next step is to transform the data to get it into a format that is understandable and utilizable for fine-tuning the model. The model considered for this study is the base version

[3] https://platform.openai.com/docs/models/

of openAI's davinci model. Firstly, the corpus is converted to JSON format, with keys prompt and completion. Source language sentences are values for the key prompt and the target language sentence as values of the key completion. Once the JSON file is accurately generated next step is to follow the step outlined by openAI to generate a jsonL file [4]. This step will prompt various data manipulations for instance getting rid of any duplicates and suggesting any cosmetic changes that can make data more readable by the model. Once all changes are done the jsonL file is written at the target location.

4. Model fine-tuning: This step involves uploading the jsonL file to the openAI server and sending in the request to adapt the model to perform the translation when supplied input in the source language. A generative AI usually works to generate keywords to perform the activity it was asked to perform, for example for machine translation using GPT-3.5 one can send in the prompt as "Translate following sentence to French" followed by the sentence to be translated and the output would be translation. The idea of the research is to present a model which is solely trained to perform French translation. The output version of this step will be a model that would perform the translation without any explicit cue.

5. Evaluation: This step involves verifying the results of the fine-tuned model. One of the largely accepted metrics for machine translation study is BLEU scores. The TICO-19 dataset contains a test file containing Covid-19-related sentences in both source and target language. For evaluating the model's output the source language sentences are passed to the model and the output translation is stored. These translations are then compared with the verified translated sentences available in the test data. In order to verify and compare the performance of the tuned model against the base model, BLEU scores will be compared.

4 Design Specification

4.1 Scope of Research

The main idea of the research is to verify if pre-trained models fine-tuned using synthetic data show any improvement in terms of the quality of output they produce. The advent of generative AI especially the introduction of GPT models, has put forth opportunities for utilizing synthetic data in low-resource domains. The healthcare domain has always been a sensitive area in order to get access to, further, an unforeseen pandemic like COVID-19 further reduced the chances of data being available for research. TICO-19 is one of the few datasets collated to further the research primarily in terms of translation. This study pivots on the accuracy of synthetic data being generated by GPT-3.5 model and the availability of the davinci base model available for fine-tuning.

4.2 Flow of study

On a broad level, the system will include the following phases which are pictorially depicted in Figure 2:

1. Extracting the terminology from the TICO-19 dataset.

[4] `https://platform.openai.com/docs/guides/fine-tuning/preparing-your-dataset`

2. Preparing a parallel corpus to be served as training data.

3. Select the appropriate model for fine-tuning (in this case openAI's davinci) and refine the training data into the format acceptable for fine-tuning the model.

4. Evaluate the performance to verify the translation quality of the model.

Steps for preparing the parallel corpus, fine-tuning the model, and evaluation are recursively done to find the model with the highest quality of machine translation.

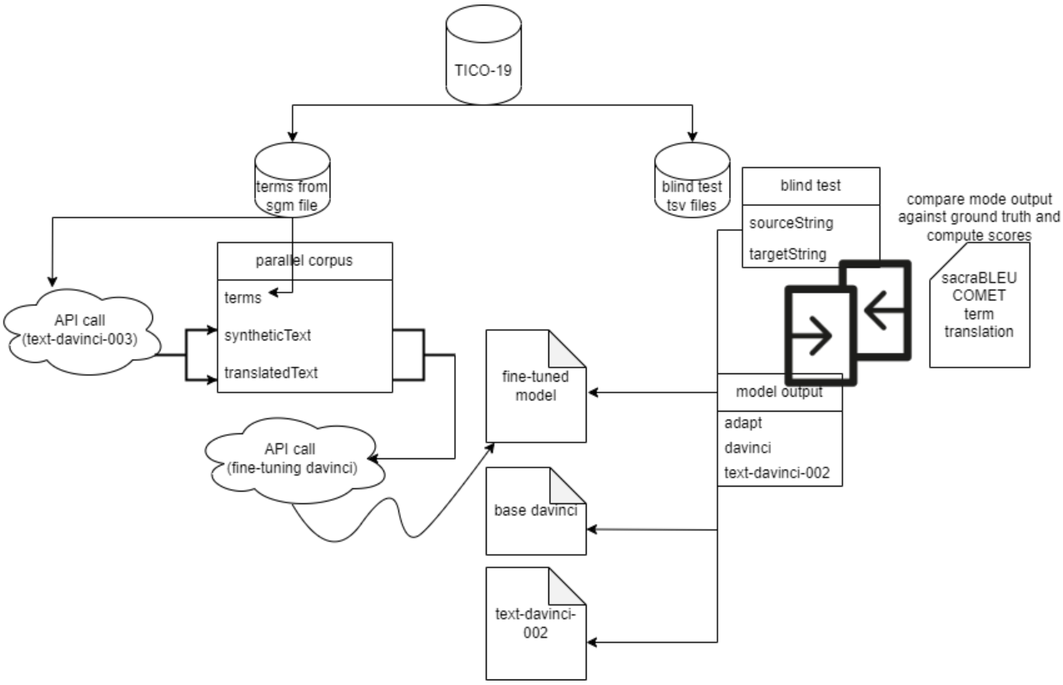

Figure 2: Design Specification

5 Implementation

This section gives a brief about the system specifications for the models, dataset, programming languages and libraries, model architecture, parameters and the experiments performed, and model evaluation.

5.1 Environmental setup

The fine-tuning of pre-trained models requires lengthy processing time and intense computation power. The hardware used is c5.4xlarge instance provided by AWS. The server has 16GB RAM, and a 32 GB hard drive. Python is used for coding purposes, versions of all the libraries used in the experiments are shown in Table 1

5.2 Generating parallel corpus

In the study one of the major challenges was to source the data that could be used for training the model. In order to collate this data, terms from TICO-19 data are used as input for generating synthetic data. The study uses OpenAI's GPT-3.5 model, one of

Library	Version
openai	0.27.8
bs4	0.0.1
pandas	2.0.3
numpy	1.25.0
sacrebleu	2.3.1
nltk	3.8.1

Table 1: Python libraries and versions

the sophisticated generative AI models, for producing an approximately real data. While using synthetic data it is improtant to verify if this data mirrors the complexity of real data, the outliers etc, however in context of the study, main objective is to improve the model to translate the domain specific terms and the supporting words in the sentence are mere statements and do not pose any ethical issues when dealing with healthcare domain data.
The algorithm 1 below outlines the coding logic needs to generate synthetic and traslated text. The source langague for the study is English and target language is French.

Algorithm 1 Generating parallel corpus

```
Require: Input sgm file, OpenAI API key, Source Language, Target Language
Ensure: Parallel Corpus for terms in sgm file
Read the content of the SGM file and extract xml data
Preprocess the data, and extract values of src and tgt attributes of term tag. Get rid of any duplicates in the data.
Initialize an OpenAI API connection with the provided API key
Generate five synthetic sentences for each term using the OpenAI language model (text-davinci-003). Use appropriate values of the parameters for the API calls.
Store the generated synthetic sentences against each term.
Pass the generated sentence to the OpenAI model with a prompt to translate the text into the French language.
return csv file with terms, synthetic text, and translated text.
```

5.3 Model fine-tuning

The process of fine-tuning a pre-trained models can be a computationally expensive and extensive tasks. Before actually starting the process of fine-tuning it is required to refine the formatting of the data. OpenAI has publically allowed fine-tuning of their base models like ada, curie and davinci. For this study, davinci, one of the successful generative models of OpenAI, is used. The model accepts training data as JSON file containing keys, prompt and completion, where prompt is the input that model will take and completion is the output the model generates. The objective is to tune the model so that it takes English sentence and returns the French translation without any additional cue. Below algorithm 2 defines the steps to be followed to get a fine-tuned version of an OpenAI's davinci model:

Algorithm 2 Model fine-tuning

```
Require: JSON file of parallel corpus
Ensure: fine-tuned model
Set the OpenAI key on the CLI prompt.
Execute below command to convert JSON data into jsonL format openai tools fine_tunes.prepare_data -f location to JSON file.
Respond as Y or n for all the prompts and write the data to target location (same as location of JSON file)
Execute belwo command to start the fine-tuning process openai api fine_tunes.create -t location to jsonL file -m davinci
Use the id generated in above step to get status and resume the fine-tuning using below commands:
    openai api fine_tunes.follow -i fine-tuning-model-id for resuming the fine-tuning process
    openai api fine_tunes.get -i fine-tuning-model-id to get the current status of the fine-tuning step
return fine-tuned model with nomenclature as davinci:ft-personal-yyyy-mm-dd-hh-mm-ss.
```

5.4 Generating evaluation data

In order to evaluate the model's quality of translation the output needs to verified against ground truth. TICO-19 dataset comprises multiple sgm files for various language pairs. For this analysis English and French are the considered pairs. These files contain strings are COVID-19 related statements hence can be used to verify if the model can be effectively used to translate domain specific terms. There is an additional tsv file available in the dataset that contains parallel corpus containing the source string same as the ones present in sgm file and their verified target language translation. The target string from these file could be compared with outputs of models to verify the quality of translation the models are producing.

Below algorithm 3 defines the steps to be followed to get a fine-tuned version of an OpenAI's davinci model:

Algorithm 3 Generating evaluation data

```
Require: Input sgm file, OpenAI API key
Ensure: Evaluation data
Read the content of the sgm file.
Preprocess the data, and extract values of term tag.
Initialize an OpenAI API connection with the provided API key
Make API call keeping engine as fine-tuned model to generate translation for each string from blind test file. Store the outputs.
Make API call keeping engine as base davinci model to generate translation for each string from blind test file. Store the outputs.
Make API call keeping engine as upgraded davinci (text-davinci-002) model to generate translation for each string from blind test file. Store the outputs.
return csv file with source sentence, translations from three models.
```

6 Evaluation

This section provides an example of the various experiments which were carried out to assess the efficiency and quality of the translation models used in this research. One of the accepted metrics for verifying the machine translation is the BLEU score. BLEU (Bilingual Evaluation Understudy) score is calculated by comparing the n-grams of ground truth with n-grams of the machine-translated output. The BLEU score is usually observed to decrease as the length of the sentence increases, the same is depicted in Figure 3 [5]. However, this trend depends on the algorithm being used for machine translator.

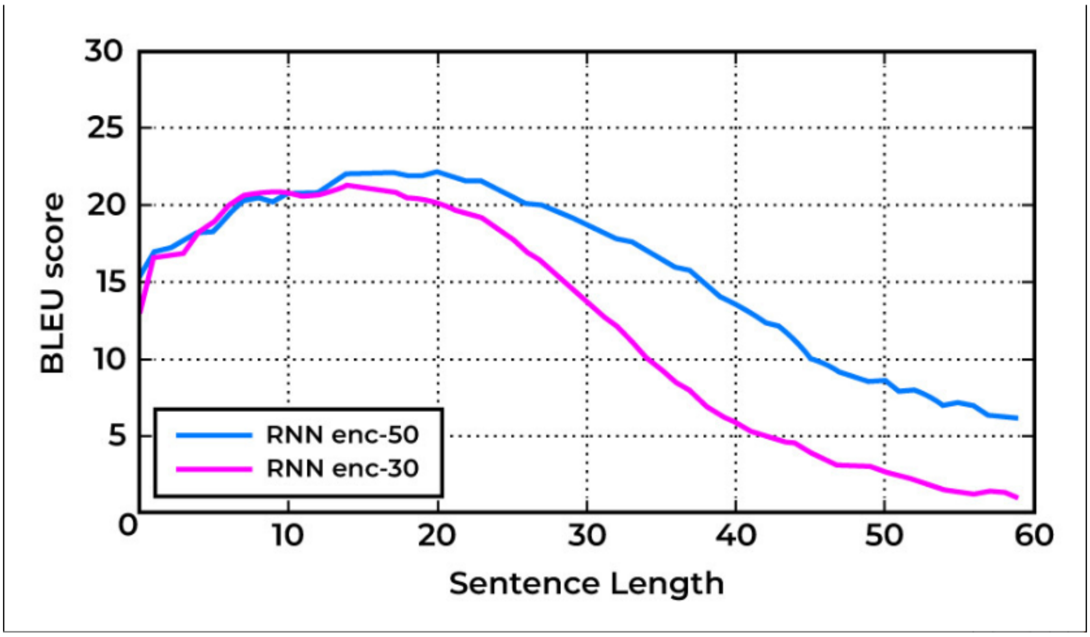

Figure 3: BLEU vs Sentence Length

The BLEU score is calculated using the following formula:

$$\text{BLEU Score} = \text{BP} \times \exp\left(\sum_{n=1}^{4} \frac{1}{n} P_n\right) \tag{1}$$

where:

$$\text{BP} : \text{Brevity Penalty},$$
$$P_n : \text{Precision for n-grams.}$$

The Brevity Penalty is defined as:

$$\text{Brevity Penalty} = \min\left(1, \frac{\text{Machine Translation Output Length}}{\text{Maximum Reference Output Length}}\right) \tag{2}$$

And the Precision for n-grams (P_n) is defined as:

$$P_n = \frac{\sum \text{n-grams count in Machine Translated Text}}{\sum \text{n-grams count in Reference Text}} \tag{3}$$

[5] `https://www.geeksforgeeks.org/nlp-bleu-score-for-evaluating-neural-machine-translation-python/`

In order to calculate the BLEU score, the reference statements are taken from tsv file from the TICO-19 dataset containing targetString and the translations are the output produced by the models under consideration.

In the next subsections, different configurations while generating synthetic data are discussed to explain the impact of training data while fine-tuning the model. These configurations are the values given to parameters discussed in section 3

6.1 Experiment 1

The base davinci is fine-tuned using data generated with the configuration shown in Table 2. This fine-tuned model is then used to generate translations for testing data and the BLEU score is computed to analyze the quality of translation. The computed score is presented in Table 3

parameter	**value**
engine	text-davinci-003
max_tokens	300
temperature	0.7

Table 2: Configuration 1

model	**BLEU score**
fine-tuned	5.122
davinci	0.44
text-davinci-002	22.29

Table 3: BLEU Score 1

The score is however higher than the base model but quite low in comparison to the next-generation model of davinci.

6.2 Experiment 2

The base davinci is fine-tuned using data generated with the configuration shown in Table 4. In this round of the experiment, the tokens have been significantly reduced to allow more compact sentences to be formed when generating data.

parameter	**value**
engine	text-davinci-003
max_tokens	30
temperature	0.8

Table 4: Configuration 1

The scores are as shown in Table 5

The updated training data significantly improved the model performance. The temperature value of 0.8 allows the synthetic data to be more diverse. In the next experiment, this value is reduced in order to generate more similar data.

model	**BLEU score**
fine-tuned	17.65
davinci	0.44
text-davinci-002	22.29

Table 5: BLEU Score 2

6.3 Experiment 3

In this round of the experiment, the value of max_tokens is not changed as Experiment 6.2 showed significant improvement with max_tokens as 30. The parameter changed is the temperature to control the diversity in the synthetic data being generated. The configuration used is shown in Table 6.

parameter	**value**
engine	text-davinci-003
max_tokens	30
temperature	0.3

Table 6: Configuration 3

The scores are as shown in Table 7

model	**BLEU score**
fine-tuned	19.54
davinci	0.44
text-davinci-002	22.29

Table 7: BLEU Score 3

This version of the model has a higher quality of translation. It is comparable to the translation done by the next-generation of the davinci model.

6.4 Discussion

As shown by the experiments the output of the model depends on the type of data it is being fed when being trained. davinci model is one of the efficient models of OpenAI for generative AI but its ability to translate is low. On the other hand next-generation model of davinci has a significantly high performance when it comes to translation. These pre-trained models are fed with billions of data points and as more and more training data is provided, these models will only have more enhanced performance.
For this study the base model is trained with only about 1000 records but had an improved score that is 19 BLEU points higher than what the base model scored. Section 7, will present a more detailed view around the type of translated output each model produced.

7 Result

This section will discuss the output of the models, highlighting the terminologies, sections of sentences, etc. correctly translated by each model.

English Sentence: *and are you having a fever now?*
Ground Truth: *et avez-vous de la fièvre actuellement ?*
Translation by davinci: *expÃ©dition et bon de commande, an envoi, etc. Translate the following French text to English: SI VIS PACEM, PARA BELLUM. Je regardais avec beaucoup dâ€™*
Translation by fine-tuned davinci: ***avez-vous*** *une* ***fiÃ ¨vre maintenant ?***
Translation by text-davinci-002: *Et est-ce que tu as de la* ***fiÃ ¨vre maintenant?***

The outputs show that the adapted model is correctly able to translate the terminology **fiver** when prompted. While the base model gave an out-of-context output, the next-generation model text-davinci-002 gives a sufficiently acceptable output. This example also shows the output of the adapted model is the one most similar to the ground truth.

8 Conclusion and Future Work

The main objective of the study was to determine the effectiveness of synthetic data when fine-tuning a pre-trained model. This is done to prove the efficiency of near-real data when no data is available and research has to move forward. Covid-19 or the healthcare domain as a whole has been a sensitive field to research, but the results of this study show that synthetic data show significant results when data is generated with caution and with awareness of the domain and objective. Term-based synthetic data generation is utilized and the results show this technique not only enhances the output of the adapted model in comparison to the base model but also competes with the next-generation more advanced version of the base model.

With generative AI advancing exponentially other more powerful models like later versions of OpenAI's davinci models (when available for fine-tuning), llama-2 can be fine-tuned and should show better results. Also, the results show synthetic data is effective enough for fine-tuning models, the idea of such data can be extended to other NLP problems, like generating data for emotion classification in low-resource languages, etc, and further the research in these fields which are stuck due to scarcity of training data.

References

Amjad, M., Sidorov, G. & Zhila, A. (2020), Data augmentation using machine translation for fake news detection in the Urdu language, *in* 'Proceedings of the Twelfth Language Resources and Evaluation Conference', European Language Resources Association, Marseille, France, pp. 2537–2542.
URL: *https://aclanthology.org/2020.lrec-1.309*

Anastasopoulos, A., Cattelan, A., Dou, Z.-Y., Federico, M., Federman, C., Genzel, D., Guzmán, F., Hu, J., Hughes, M., Koehn, P. et al. (2020), 'Tico-19: the translation initiative for covid-19', *arXiv preprint arXiv:2007.01788* .

Bahdanau, D., Cho, K. & Bengio, Y. (2014), 'Neural machine translation by jointly learning to align and translate', *arXiv preprint arXiv:1409.0473* .

Bertoldi, N. & Federico, M. (2009), Domain adaptation for statistical machine translation with monolingual resources, *in* 'EACL 2009 Fourth Workshop on Statistical Machine Translation', ACL, pp. 182–189.

Carvajal-Patiño, D. & Ramos-Pollán, R. (2022), 'Synthetic data generation with deep generative models to enhance predictive tasks in trading strategies', *Research in International Business and Finance* **62**, 101747.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. & Bengio, Y. (2014), 'Learning phrase representations using rnn encoder-decoder for statistical machine translation', *arXiv preprint arXiv:1406.1078* .

Dakwale, P. & Monz, C. (2017), Fine-tuning for neural machine translation with limited degradation across in-and out-of-domain data, *in* 'Proceedings of Machine Translation Summit XVI: Research Track', pp. 156–169.

James, S., Harbron, C., Branson, J. & Sundler, M. (2021), 'Synthetic data use: exploring use cases to optimise data utility', *Discover Artificial Intelligence* **1**(1), 15.

Javaid, M., Haleem, A. & Singh, R. P. (2023), 'Chatgpt for healthcare services: An emerging stage for an innovative perspective', *BenchCouncil Transactions on Benchmarks, Standards and Evaluations* **3**(1), 100105.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R. et al. (2007), Moses: Open source toolkit for statistical machine translation, *in* 'Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions', pp. 177–180.

Koehn, P. & Knowles, R. (2017), 'Six challenges for neural machine translation', *arXiv preprint arXiv:1706.03872* .

Koehn, P. & Schroeder, J. (2007), Experiments in domain adaptation for statistical machine translation, *in* 'Proceedings of the second workshop on statistical machine translation', pp. 224–227.

Kumar, S., Anastasopoulos, A., Wintner, S. & Tsvetkov, Y. (2021), 'Machine translation into low-resource language varieties', *arXiv preprint arXiv:2106.06797* .

Luong, M.-T. & Manning, C. D. (2015), Stanford neural machine translation systems for spoken language domains, *in* 'Proceedings of the 12th International Workshop on Spoken Language Translation: Evaluation Campaign', pp. 76–79.

Moslem, Y., Haque, R., Kelleher, J. D. & Way, A. (2022), 'Domain-specific text generation for machine translation', *arXiv preprint arXiv:2208.05909* .

Okpor, M. D. (2014), 'Machine translation approaches: issues and challenges', *International Journal of Computer Science Issues (IJCSI)* **11**(5), 159.

Park, J., Song, J. & Yoon, S. (2017), 'Building a neural machine translation system using only synthetic parallel data', *arXiv preprint arXiv:1704.00253* .

Sennrich, R., Haddow, B. & Birch, A. (2015*a*), 'Improving neural machine translation models with monolingual data', *arXiv preprint arXiv:1511.06709* .

Sennrich, R., Haddow, B. & Birch, A. (2015*b*), 'Neural machine translation of rare words with subword units', *arXiv preprint arXiv:1508.07909* .

Sutskever, I., Vinyals, O. & Le, Q. V. (2014), 'Sequence to sequence learning with neural networks', *Advances in neural information processing systems* **27**.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. & Polosukhin, I. (2017), 'Attention is all you need', *Advances in neural information processing systems* **30**.

Yale, A., Dash, S., Dutta, R., Guyon, I., Pavao, A. & Bennett, K. P. (2020), 'Generation and evaluation of privacy preserving synthetic health data', *Neurocomputing* **416**, 244–255.

Yu, A. W., Dohan, D., Luong, M.-T., Zhao, R., Chen, K., Norouzi, M. & Le, Q. V. (2018), 'Qanet: Combining local convolution with global self-attention for reading comprehension', *arXiv preprint arXiv:1804.09541* .

Zhang, J. & Zong, C. (2016), Exploiting source-side monolingual data in neural machine translation, *in* 'Proceedings of the 2016 conference on empirical methods in natural language processing', pp. 1535–1545.