Machine Translation

Name: Bongwoo Jeon

I. ABSTRACT

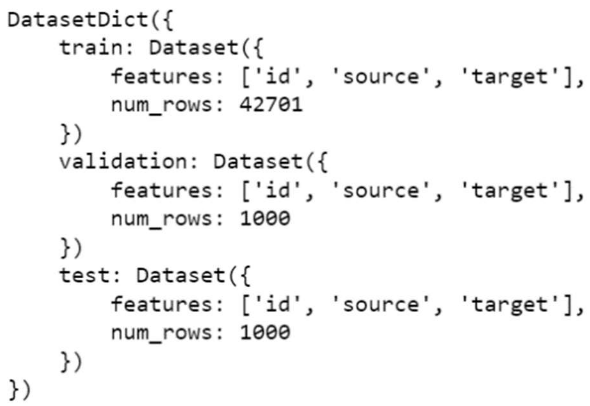
Throughout this course, I was introduced to many machine learning techniques for use in text classification problems. In this paper, I set out to apply these methods in the analysis of sample data sets provided by Hugging Face. I compare the training loss, learning rate, BLEU scores of a variety of prediction methods on the CodeXGLUE text-to-text data set, namely the MT5Model. I compare the performance of Different language pairs in this dataset.

II. INTRODUCTION

My goal of this final project is to create a simple translation based on MT5 model and understand how this model works and evaluate all language pairs in the dataset.

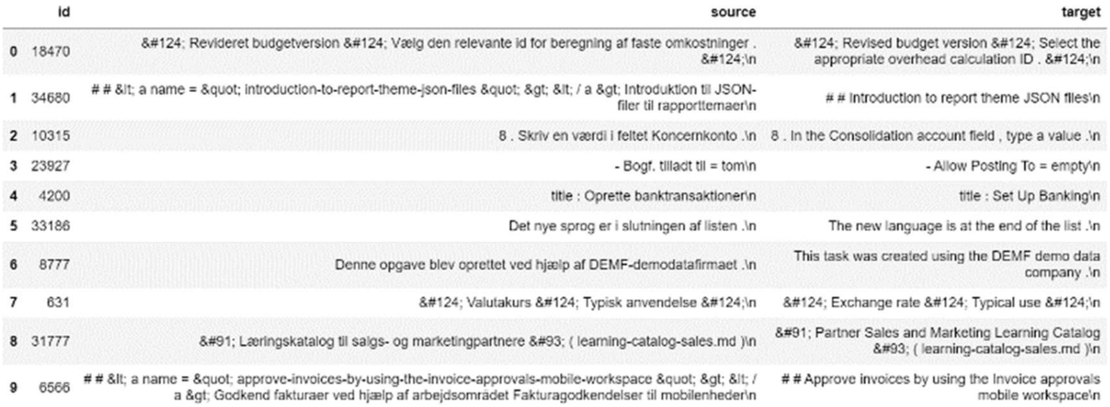
III. DATASET

My data set I selected was translation data set including four languages which are Danish-English, Latvian-English, Norwegian-English, Chinese-English. This data set, originally from the Microsoft Documentation, consists of 155,926 train data, 4,000 validation data, 4,000 test data. So total 163,926 instances. The attributes set contains the id, source, target. “source” column contains Danish, Latvian, Norwegian, and Chinese texts. “target” column contains only English texts. So, Fig 1 shows that how the dataset is configured. For example, the first step I took in working on a new data set is to explore it. I used a series of plots to visualize and understand the data better. I attached 10 random samples of Danish-English dataset picture which contained ‘id’, ‘source’, ‘target’ columns. ‘source’ column is Danish texts, and ‘target’ column is English texts like Fig 2.



```
DatasetDict({
    train: Dataset({
        features: ['id', 'source', 'target'],
        num_rows: 42701
    })
    validation: Dataset({
        features: ['id', 'source', 'target'],
        num_rows: 1000
    })
    test: Dataset({
        features: ['id', 'source', 'target'],
        num_rows: 1000
    })
})
```

Fig.1. Dataset Configuration



	id	source	target
0	18470	| Revideret budgetversion | Vælg den relevante id for beregning af faste omkostninger . |\n	| Revised budget version | Select the appropriate overhead calculation ID . |\n
1	34680	# # &lt; a name = &quot; introduction-to-report-theme-json-files &quot; &gt; &lt; / a &gt; Introduktion til JSON-filer til rapporttemaer\n	# # Introduction to report theme JSON files\n
2	10315	8 . Skriv en værdi i feltet Koncernkonto .\n	8 . In the Consolidation account field , type a value .\n
3	23927	- Bogf. tilladt til = tom\n	- Allow Posting To = empty\n
4	4200	title : Oprette banktransaktioner\n	title : Set Up Banking\n
5	33186	Det nye sprog er i slutningen af listen .\n	The new language is at the end of the list .\n
6	8777	Denne opgave blev oprettet ved hjælp af DEMF-demodatafirmaet .\n	This task was created using the DEMF demo data company .\n
7	631	| Valutakurs | Typisk anvendelse |\n	| Exchange rate | Typical use |\n
8	31777	&#91; Læringskatalog til salgs- og marketingpartnere &#93; (learning-catalog-sales.md)\n	&#91; Partner Sales and Marketing Learning Catalog &#93; (learning-catalog-sales.md)\n
9	6566	# # &lt; a name = &quot; approve-invoices-by-using-the-invoice-approvals-mobile-workspace &quot; &gt; &lt; / a &gt; Godkend fakturaer ved hjælp af arbejdsområdet Fakturagodkendelser til mobilenheder\n	# # Approve invoices by using the Invoice approvals mobile workspace\n

Fig.2. 10 samples of Danish-English dataset

IV. Model

I used MT5Model(mt5-small) which is supported from Google to train dataset. mT5 is a multilingual Transformer model pre-trained on a dataset (mC4) containing text from 101 different languages. The architecture of the mT5 model (based on T5) is designed to support any Natural Language Processing task (classification, NER, question answering, etc.) MT5Model(mt5-small) is consist of 8 encoder and 8 decoder layers, and it is fully pre-trained seq2seq network. It has already both pre-trained encoder and decoder layers. So, I didn't have to create and add additional pre-trained component separately. I added Fig 3 that the mT5 model is configured.

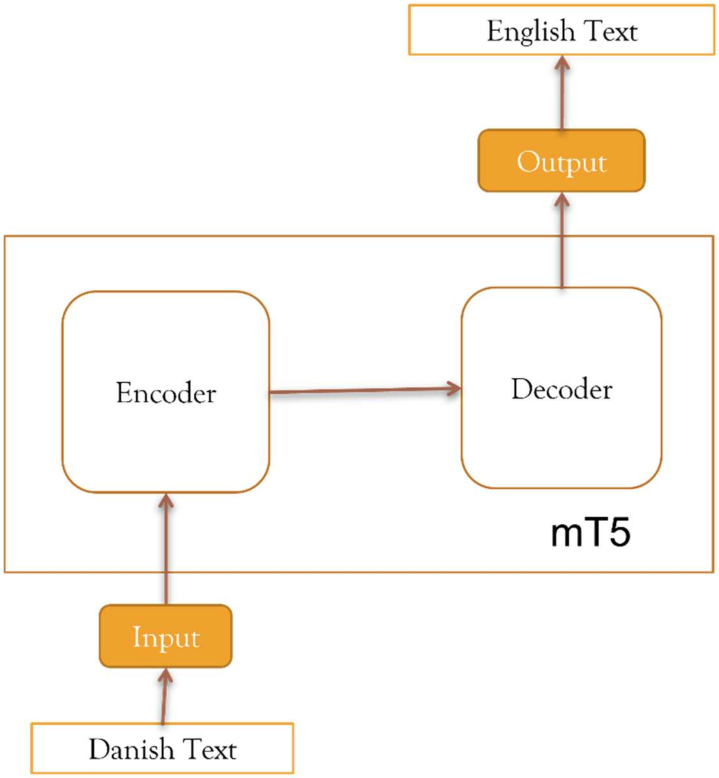


Fig.3. MT5Model

V. Experimental Result

First, I set 8 batch size, 1 number of train epoch, 5e-e learning rate, 128 max sequence length for hyperparameters to train a model and evaluate metric scores for all language pairs. I got the final learning rate 3.595e-7 for Danish-English, 9.479e-7 for Latvian-English, 3.737e-7 for Norwegian-English, and 1.611e-7 for Chinese-English. Also, I got training loss 0.03138 for Danish-English, 0.6086 for Latvian-English, 0.01285 for Norwegian-English, and 0.03115 for Chinese-English. And I got the BLEU scores 54.55% for Danish-English, 63.27% for Latvian-English, 54.58% for Norwegian-English, and 56.72% for Chinese-English.

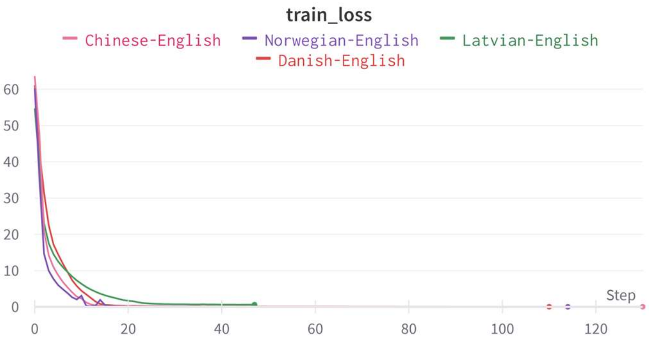


Fig 4. Train loss for all language pairs

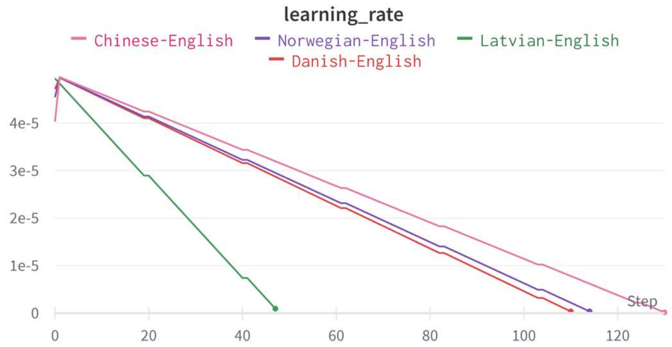


Fig 5. Learning rate for all language pairs

	Danish-English	Latvian-English	Norwegian-English	Chinese-English
BLEU	54.55	63.24	54.58	56.72

Table 1. BELU Comparison

VI. Analysis

Generally, a large learning rate allows the model to learn faster, at the cost of arriving on a sub-optimal final set of weights. A smaller learning rate may allow the model to learn a more optimal or even globally optimal set of weights but may take significantly longer to train. So, as you can see the Fig 5. mT5 model learns quickly in the order of Danish-English, Latvian-English, Norwegian-English, and Chinese-English. Loss can be seen as a distance between the true values of the problem and the values predicted by the model. Greater the loss is, huger is the errors you made on the data. Accuracy can be seen as the number of errors you made on the data. a low accuracy and huge loss mean value I made huge errors on a lot of data. According to Fig 4, model has high accuracy in the order of Norwegian-English, Chinese-English, Danish-English, and Latvian-English. BLEU (Bilingual Evaluation Understudy) is a metric for automatically evaluating machine-translated text. The BLEU score is a number between zero and one that measures the similarity of the machine-translated text to a set of high-quality reference translations. If BLUE score is 50~60%, that means quality is very high, adequate, and fluent translations. And if BLUE score is up to 60, this means Quality is often better than human. So, in the Table 1, Danish-English, Norwegian-English, Chinese-English got scores between 50~60%. But only Latvian-English got score up to 60%. That means Danish-English, Norwegian-English, Chinese-English is very high quality but not fluent than human. But Latvian-English's quality is often better than human.

VII. Conclusion

My mT5 model learned quickly in the order of Danish-English, Latvian-English, Norwegian-English, and Chinese-English in terms of learning rate. And model had high accuracy in the order of Norwegian-English, Chinese-English, Danish-English, and Latvian-English in terms of training loss. In terms of BLEU scores, Danish-English, Norwegian-English, Chinese-English is very high quality but not fluent than human. But Latvian-English's quality is often better than human. The ranking of learning rate, loss rate, and BLUE scores were different for each language pair. Therefore, we cannot conclude that any of the four language pairs are exceptionally superior.

VIII. FUTURE WORK

I used only MT5-small model for this project this time. There are different types of MT5 model which are MT5-small, MT5-base, MT5-large, MT5-xl, MT5-xxl. So I couldn't train and evaluate and compare all these models, and I couldn't determine which model will be best for my current dataset. And also, I tried to make some beautiful demo using gradio or streamlit app for my model, but I couldn't complete to make this demo this time. I want to make very beautiful demo and train different MT5 models if I have an enough time later.