TRANSLATING ENGLISH TO NORWEGIAN USING MULTILINGUAL LANGUAGE MODELS

Jørgen Navjord

NMBU, Norway
Email: {jorgenav@nmbu.no}

ABSTRACT

In this paper, I investigate the use of T5, a family of language models for translating natural language from English to Norwegian. The motivation for this research is to determine whether it is feasible to translate English summarization datasets to Norwegian for our master thesis. I then begin by describing the theory behind the T5 model and its components. Then I fine-tune three pre-trained T5 models, called Flan-T5, on a translation task using sentence pairs extracted from the Bible. Our results show that the best-performing model achieves a BLEU score of 36.7, which is comparable to the state-of-the-art on other translation benchmarks. Based on these results, I conclude that it is realistic to use automatic translations for translating summarization datasets to Norwegian.

Index Terms— deep learning, machine learning, transfer learning, translation, neural machine translation, T5

1. INTRODUCTION

The following report will explore if it is possible to use a state-of-the-art transformer model to translate English to Norwegian. The motivation for the topic is to explore the possibilities of using English summarization datasets that have been translated into Norwegian for automatic text summarization of Norwegian text. The thesis will be written in the spring of 2023 on an assignment from Skatteetaten (The Norwegian Tax Administration). They are particularly interested in exploring how natural language processing (NLP) can be used to make advanced tax documentation more accessible for both internal and external users of their services.

The task of natural text translation is one of the most studied topics in linguistics and machine learning. The task aims to map from a source language to a target language automatically.

For a long time, this was performed using Statistical Machine Translation (SMT) [1]. This set of models leverages probabilistic models to capture similarities between pair-wise sentences. Often, these models also required hand-made features which are difficult and require domain knowledge. Google Translate is an example of a service which used SMTs for a long time.

In 2016, Google announced that they had started using a set of translation models called Neural Machine Translation (NMT) models which outperformed their older statistical models [2]. NMT models are a set of models using deep learning for translation. It was first proposed in ”Recurrent Continuous Translation Models” by Kalasdsad et al [3]. The paper described how a subset of deep learning models called Recurrent Neural Networks (RNNs) was able to achieve state-of-the-art results on various translation tasks. These models use an encoder with recurrent units with a hidden state capable of picking up on the most relevant information and sending it to a latent space. The decoder then decodes the vector back into tokens. A different recurrent unit called Long Short-Term Memory (LSTM) is more capable of creating meaningful representations in the latent space have also been used for NMT and was responsible for the state-of-the-art solutions for many years [4].

The use of transformer models has further revolutionized the field of neural machine translation. Unlike previous models, which relied on RNNs, transformers are based on self-attention mechanisms, allowing them to capture long-range dependencies in the input data [5]. This has allowed them to outperform previous state-of-the-art models on many translation tasks. Additionally, transformer models have the advantage of being more efficient to train and faster to run, making them more practical for use in real-world applications. Today, Many of the most popular translation services, such as Google Translate, now use transformer models to power their translations [6].

Despite neural machine translation being a well-studied problem in the field of natural language processing, with many papers and research articles published on the topic. However, to the best of my knowledge, there have not been any published papers specifically focused on translating English to Norwegian using NMT. This might be because of a lack of public parallel corpora or other resources available for training and evaluating NMT systems for English-Norwegian translation. Despite these challenges, there do exist commercial offerings such as Google Translate which have been able to provide translations on demand between the two languages since 2008 [7].

2. THEORY

The chapter of this paper covers several important concepts and techniques in natural language processing, including transfer learning and fine-tuning, tokenization, language models, masked language modeling, the attention mechanism, transformer models, and the BLEU metric.

2.1. Transfer learning and fine tuning

Fine-tuning in machine learning is a method of training a neural network by making small adjustments to the weights of the network to improve its performance on a specific task. This is typically done after the network has already been trained on a large dataset for a general purpose, and the goal of fine-tuning is to make the network more adaptable to the specific task it will be used for. This can save time and computational resources, and can also improve performance by taking advantage of the knowledge already learned by the pre-trained network. Fine-tuning is a widely used technique in deep learning and has been shown to be effective in many applications, such as image classification and natural language processing.

2.2. Tokenizer

A tokenizer is a program or function that takes a string of text as input and divides it into a sequence of tokens. A token is a unit of text that is meaningful to the program being used, such as a word, number, punctuation mark, or other string of characters.

Tokenization is the process of breaking down a string of text into these individual tokens. It is a crucial step in natural language processing tasks such as text analysis and language modeling, as it allows the system to operate on individual words and punctuation marks rather than on the raw text as a whole.

Tokenizers typically follow a set of rules or algorithms to determine how to divide the input text into tokens. For example, a tokenizer might split a string of text into words by using whitespace and punctuation marks as delimiters. It might also use specific rules for handling numbers, abbreviations, and other special cases.

2.2.1. SentencePiece

SentencePiece is an open-source tokenizer and detokenizer for natural language processing tasks. It is a popular choice for tokenization in NLP systems, especially for neural machine translation. It is developed by Google and was first described in ”SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing” by Kudo, Taku, and John Richardson [8].

SentencePiece is designed to be flexible and efficient, with features such as:

- Support for subword segmentation and encoding, which allows the tokenizer to handle rare and out-of-vocabulary words effectively.
- simple, trainable, and language-agnostic design that allows it to be easily adapted to different languages and domains.
- stable and fast implementation that can handle large amounts of text in real-time.

To use SentencePiece, you first need to train it on a corpus of text in the target language. This process involves specifying the desired vocabulary size and other parameters The resulting SentencePiece tokenizer can then be used to tokenize and detokenize text as needed.

2.3. Language Models

A language model is a probabilistic model that assigns a likelihood to a given sequence of words [9]. Formally, let $w_1, w_2, \ldots, w_T$ be a sequence of words, where T is the length of the sequence. The language model assigns a probability to this sequence as follows:

$$P(w_1, w_2, \ldots, w_T) = \prod_{t=1}^{T} P(w_t \mid w_1, w_2, \ldots, w_{t-1})$$

In other words, the probability of a sequence of words is the product of the probabilities of each word given the preceding words. This formulation captures the idea that the likelihood of a word depends on the context in which it appears.

Language models are typically trained on large amounts of text data, such as books, articles, and other written documents. The goal of training a language model is to learn the underlying structure and patterns of the language in order to make more accurate predictions about the likelihood of a given sequence of words.

Language models are used in a variety of natural language processing tasks, including language generation, language translation, and text summarization. They are also used in more specialized tasks, such as dialogue systems and sentiment analysis. Overall, language models are an important tool for understanding and generating natural language text.

2.4. Masked Language Modelling

Masked language modeling (MLM) is a type of language modeling task in which some words in the input are randomly masked, and the model is trained to predict the masked words based on the context provided by the rest of the input.

For example, given the input sentence ”The quick brown fox jumps over the lazy dog,” an MLM training procedure might mask the words ”quick” and ”brown” and train to predict their original values. In this case, the model would be trained to output the sequence ”The [MASK] [MASK] fox jumps over the lazy dog,” where the masked tokens are predicted by the model.

Masked language modeling has been shown to improve the performance of language models on a variety of tasks, including machine translation and text summarization. It has become a key component of many state-of-the-art models, such as the BERT model introduced by Devlin et al. in their 2018 paper ”BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding” [10].

In BERT, MLM is used as the pre-training task to help the model learn the general structure of natural language. This pre-training step is important because it allows the model to learn useful representations of the input that can be fine-tuned for specific downstream tasks, such as sentiment analysis or question answering.

2.5. Attention and self-attention

Attention mechanisms have become an important part of many natural language processing models. Attention mechanisms allow models to focus on specific parts of an input, rather than processing the entire input equally. This can make the model more efficient and improve its performance.

Self-attention, in particular, allows the model to attend to different parts of the input sequence and compute their relevance to each other, without requiring any additional input or context. This is done using a combination of matrix multiplication and softmax normalization, as shown in the following equation:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Here, Q, K, and V are matrices representing the query, key, and value for each element in the input sequence, respectively. d_k is the dimensionality of the keys. The output of the self-attention layer is a weighted sum of the values, where the weights are determined by the dot product of the query and the key for each element.

2.6. The Transformer

The Transformer model is a neural network architecture that was introduced in 2017 by Vaswani et al [5] in their paper ”Attention is All You Need.” This model uses self-attention mechanisms to process input sequences in parallel, allowing it to perform sequence-to-sequence tasks such as machine translation more efficiently than previous models such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs).

The Transformer model consists of two main components: an encoder and a decoder (See Fig 1 for illustration). The encoder processes the input sequence and produces a fixed-length representation of the input, which is then passed to the decoder. The decoder uses this representation to generate the output sequence. Both the encoder and decoder are made up of multiple layers of self-attention and fully connected layers.

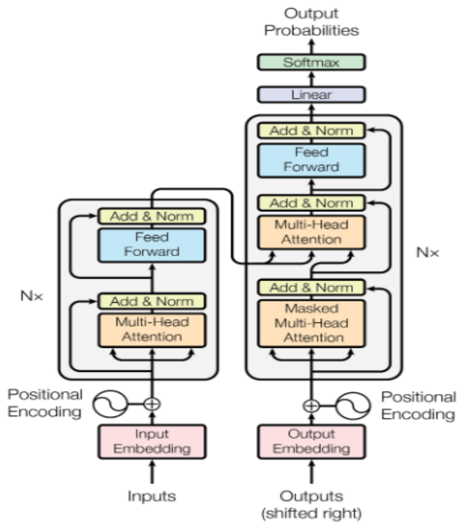

Fig. 1: Illustration of the encoder-decoder transformer model [5].

One key advantage of the Transformer model over RNNs and CNNs is its ability to process input sequences in parallel, using self-attention mechanisms instead of sequential processing. This allows the model to scale to longer sequences and achieve better performance on tasks such as machine translation.

One potential drawback of the attention mechanism is that it can be computationally expensive, especially for large inputs. Because the attention mechanism allows the model to weigh and combine information from all input tokens, the model must compute the attention weights for every possible input token pair, which can be quadratic in the number of input tokens. This can make transformers slower to train and use than some other models. Additionally, the attention mechanism can sometimes cause the model to focus too heavily on certain input tokens at the expense of others, potentially leading to suboptimal performance.

Many state-of-the-art machine learning models in NLP and Computer Vision are based on the Transformer architecture. Some examples of famous models based on Transformers include the BERT model developed by Google, the GPT-models developed by OpenAI and The Vision Transformer (ViT) [10–12].

2.6.1. T5

The T5 models are a set of models introduced in the paper ”Exploring the Limits of Transfer Learning with a Unified

Text-to-Text Transformer" by Colin, Raffel, et al [13]. The models are based on the Transformer architecture that incorporates several modifications to the original architecture, including a relative position representation and a reversible encoder-decoder layer.

One key difference between the Transformer and the T5 model is the way they are trained and used. The Transformer is typically trained on a specific NLP task, such as machine translation, and is optimized for that task. In contrast, the T5 model is trained on a wide range of NLP tasks using a unified text-to-text format for the input and output, allowing it to perform multiple tasks using a single model architecture and training procedure (See Fig. 4 for examples of the tasks).

Another difference is the performance of the two models on NLP tasks. The T5 model has been shown to achieve state-of-the-art performance on a wide range of NLP tasks, including text classification, summarization, translation, and question-answering. In contrast, the Transformer model has been shown to perform well on some NLP tasks, but not as well as the T5 model on the full range of tasks.

2.7. BLEU

BLEU (Bilingual Evaluation Understudy) is a metric for evaluating the quality of machine translation introduced in "Bleu: a method for automatic evaluation of machine translation." by Papineni, Kishore, et al. in 2002 [14]. BLEU is based on the idea that a good translation should have a high degree of n-gram overlap with a reference translation. BLEU scores can range from 0 to 100, with higher scores indicating better performance. However, in practice, most models will not achieve a BLEU score of 100, and scores in the range of 20-40 are more common.

The BLEU score is calculated as follows:

$$BLEU = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} \frac{\log p_n}{N}\right)$$

Here, N is the maximum order of the n-grams considered (commonly $N = 4$), p_n is the precision of the translation with respect to the reference translation at n-gram order n, and the brevity penalty is a function of the translation length.

The precision p_n is calculated as:

$$p_n = \frac{\sum_{\text{ngram}\in\text{candidate}} \text{count}_{\text{clip}}(\text{ngram})}{\sum \text{ngram} \in \text{candidatecount}(\text{ngram})}$$

Here, count(ngram) is the number of times the n-gram appears in the candidate translation, and $\text{count}_{\text{clip}}(\text{ngram})$ is the number of times the n-gram appears in the reference translation, clipped to the number of times it appears in the candidate translation.

The brevity penalty (BP) is calculated as:

$$BP = \begin{cases} 1 & \text{if } c \geq r \\ exp(1 - \frac{r}{c}) & \text{if } c \leq r \end{cases}$$

Here, c is the length of the candidate translation and r is the length of the reference translation. The brevity penalty encourages the candidate translation to be as short as the reference translation up to a certain point but penalizes translations that are significantly shorter.

BLEU scores are commonly used in machine translation research to compare the performance of different translation models. A higher BLEU score indicates that the candidate translation has a higher degree of n-gram overlap with the reference translation, and is therefore considered to be of higher quality.

3. METHODS

The following chapter describes the process of identifying a suitable dataset for fine-tuning the T5 models. A few selected T5 models are presented and their characteristics are overviewed. The fine-tuning process is also described, including the software used and the hyperparameters employed.

3.1. Dataset

Multiple datasets were considered for fine-tuning the translation models (See Table 1). I had a few requirements when deciding on a dataset. The first is that the data should already be structured into pairs of English and Norwegian sentences. This makes it a lot easier to write software for training the models. The second one is that the data should already be cleaned. Many translation datasets such as the CCMatrix are based on scraped data from the internet [15]. Having to clean the HTML manually would slow down the process and is out of the scope of this project. The last requirement was that it should be available on the Hugging Face Hub. The Hugging Face Hub provides access to a wide range of pre-trained transformer models, diffusion models, and datasets for NLP tasks and it makes it easy to download the data and use it for training transformer models. From these requirements, I found three candidate datasets browsing the Hugging Face Hub and filtering by datasets for translation with sentence pairs in English and Norwegian [16].

CCMatrix is the largest of them and it includes 47.8 million sentence pairs [15]. It is made by Meta AI and contains data extracted from the CommonCrawl public dataset. The pairs of sentences are generated by calculating the distance between embeddings generated from the sentences in the data.

Bible-para is a multilingual dataset generated from the Bible. It contains sentence pairs in 102 languages, including English and Norwegian. The English and Norwegian subset has a total of 62 107 sentence pairs [17].

Table 1: The different datasets considered that included English-Norwegian sentence pairs.

Dataset name	Sentence pairs
CCMatrix	47.8M
bible-para	62 107
europa-eac-tm	9 523

Europa-eac-tm consists of sentence pairs extracted from electronic forms for the EUs EAC's Life-long Learning Programme (LLP) and the Youth in Action Programme [18]. The dataset contains 9 523 sentence pairs from English to Norwegian.

After evaluating these datasets I decided to focus on the Bible-para dataset. This was done because of the size making it more accessible for doing my project. CCMatrix has too many sentence pairs for fine-tuning on the hardware available and it would be too slow. Europa-eac-tm would be easy and efficient to fine-tune models on, but I was worried that it would be too domain-specific and not contain sentences representing all of English and Norwegian. Bible-para should be a middle ground between this by being realistic to fine-tune and not being too focused on one domain.

The next step was to split the dataset into two subsets, a train set, and a validation set. 80% of the total data was used for fine-tuning and the final 20% for validation.

3.2. Models

Since the T5 models are a family of models, there are many public pre-trained models to choose between.

3.2.1. T5

Google published 5 different pre-trained models with the number of parameters ranging from 80M to 11B when publishing their original paper in 2019 [13]. These models were trained on the c4 corpus, a cleaned version of the Common Crawl corpus in English [19]. However, these models are not suitable for translating between languages other than English, such as Norwegian due to not being trained on Norwegian tasks.

3.2.2. mT5

To address this limitation, the mT5 model was introduced in the paper ”mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer” by Xue et al. in 2021 [20]. This multilingual variant of T5 was trained on the mc4 corpus, which includes 101 languages, including English and Norwegian.

3.2.3. Flan-T5

In ”Scaling Instruction-Finetuned Language Models” by Chung, Hyung Won et al. in 2022, the Flan-T5 model was introduced [21]. This model differs from mT5 by increasing the number of training tasks, using larger model sizes, and including chain-of-thoughts as a hint in the prompts. The Flan-T5 model outperforms mT5 on a majority of tasks.

In this research, the Flan-T5 model family was chosen for fine-tuning, due to its pre-training on both English and Norwegian, and its superior performance on benchmarks compared to the original T5 model. Specifically, the Flan-T5-small, Flan-T5-base, and Flan-T5-large models with 80 million, 247 million, and 780 million parameters, respectively, were used in the experiments (See table 2).

3.3. Training task

In order to specify the task that the model should perform, a prefix is added to the source text. For instance, if the text to be translated is ”The weather is nice today,” the final prompt to the model would be ”translate English to Norwegian: The weather is nice today.” This approach is consistent with the translation examples that the T5 models have been pre-trained on.

3.4. Software and Hardware

The software used is Python 3.7 with the following modules:

- **Pytorch** (v1.13.0): PyTorch is an open-source machine learning library for Python that provides a suite of tools for developing and training deep learning models [22]. It is designed to be flexible, user-friendly, and extensible, allowing users to implement custom neural network architectures for their specific needs. PyTorch is often used in natural language processing, computer vision, and other areas of AI research, and has been adopted by many major companies and research institutions. It provides a strong foundation for developing cutting-edge machine learning models and algorithms. Pytorch was not used directly in this project, but was chosen as the machine learning backend in transformers from Hugging Face.

- **Transformers** (v4.26.0.dev0): Transformers is an open-source library from Hugging Face for natural language processing that provides tools for training and deploying state-of-the-art NLP models in a variety of languages [23]. The library is designed to be user-friendly and easy to use, with a focus on flexibility and extensibility. It includes pre-trained models for a wide range of NLP tasks, as well as tools for fine-tuning and customizing these models for specific needs. It is widely used by researchers, developers, and data scientists for a variety of NLP applications.

- **datasets** (v2.7.1): The datasets python module from Hugging Face is a library that provides access to a large number of datasets for NLP tasks. These datasets can be used to train machine learning models, such as language models and text classifiers, to perform a wide range of tasks, including text classification, sentiment analysis, and machine translation.

- **evaluate** (v0.3.0): The evaluate python module from Hugging Face is a library that provides tools for evaluating the performance of natural language processing models. It is designed to work with the datasets and transformers module to provide a unified framework for training and evaluating NLP models. The evaluate module includes a variety of metrics and evaluation functions for assessing the performance of a model on a given dataset, such as accuracy, precision, recall, and F1 score. These tools can be used to compare the performance of different models and to fine-tune model hyperparameters for optimal performance. It also provides a simple API to extend the module with custom metrics such as BLEU.

- **sacrebleu** (v2.3.1): The sacrebleu python module is a library for calculating BLEU scores. The sacrebleu module is designed to be easy to use and efficient, and it has become a standard tool for evaluating machine translation models in the natural language processing community.

Tokenization and fine-tuning was performed on hardware rented from vast.ai. Different servers were rented for fine-tuning the different models, but they all had 4 GTX 3090 GPUs available. A custom docker image built on Ubuntu 18.04 was made to ensure similar training procedures across all rented servers.

Table 2: Number of parameters and hyperparameters used for fine-tuning selected Flan-T5 models. Batch size is total batch size across all GPUs.

Model	Parameters	Batch size
Flan-T5-small	80M	256
Flan-T5-base	247M	64
Flan-T5-large	780M	16

3.5. Hyperparameters and training setup

The Adam optimizer was used for training all three models, with β_1=0.9, β_2=0.999, and $\epsilon = 10^{-8}$ [24]. A linear learning rate scheduler was employed, with an initial learning rate of 5^{-5}, and early stopping was used to prevent overfitting. The models were fine-tuned using cross-entropy loss as the objective function. The batch size was determined based on the available memory on the GPU and is shown in Table 2.

4. RESULTS

This chapter will provide an overview of the train and validation data as well as the results and computed metrics for the fine-tuned models. A more detailed analysis of the results will be covered in the discussion section.

4.1. Data split

The data was split with 80% of the data in the train set and 20% in the validation set as described in the method section. This resulted in 49 685 sentence pairs in the train set and 12 421 sentence pairs in the validation set.

An initial exploratory data analysis showed that when counting the number of characters in the English and Norwegian sentences, the average English sentence was longer than the Norwegian sentences. This was consistent across both the train and validation sets (Fig.3 and Fig.4).

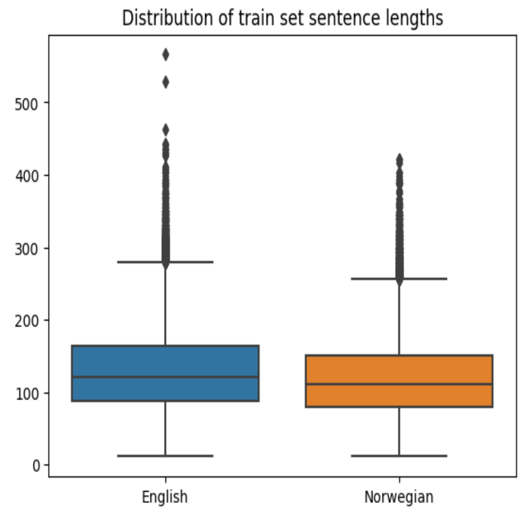

Fig. 2: Distribution of the lengths of English and Norwegian sentences in the train set.

4.2. Training results

The next step was to tokenize and fine-tune the chosen models on the datasets generated in the previous section. Using Flan-T5-small, our baseline achieved a BLEU score of 15.53. It converged after 20 epochs and was trained in total for one hour. The base model converged after 10 epochs and a total training time of 6 hours and achieved a BLEU score on the validation set of 28.72. The largest model, Flan-T5-large converged after 5 epochs and a total training time of 22 hours and achieved a BLEU score of 36.72. The findings are summarized in Table 3.

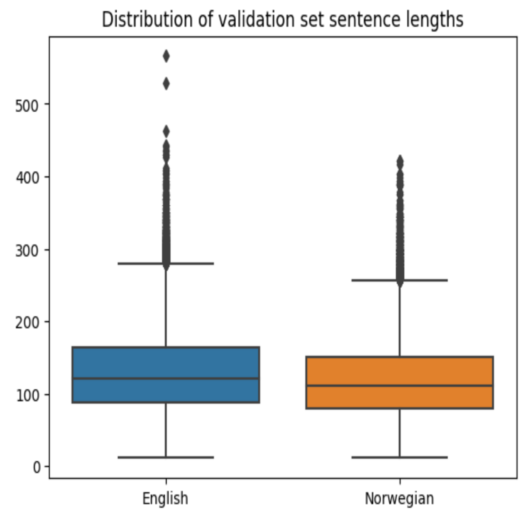

Fig. 3: Distribution of the lengths of English and Norwegian sentences in the validation set.

Table 3: Results of various pre-trained Flan-T5 models trained on bible-para to translate text from English to Norwegian evaluated on a validation set.

Model	BLEU	Epochs	Training time
Flan-T5-small	15.5285	20	1 hour
Flan-T5-base	28.7219	10	6 hours
Flan-T5-large	36.7184	5	22 hours

5. DISCUSSION

The discussion chapter of this research paper evaluates the performance of fine-tuned flan-t5 models on the English-to-Norwegian language pair. BLEU scores on the validation set are used to assess the accuracy and reliability of the models. The chapter also presents example outputs from the models on self-generated prompts, providing further insight into the behavior and quality of the models.

5.1. Metrics

There are no known benchmarks for evaluating the performance of machine translation models on the English-to-Norwegian language pair. This lack of benchmarks makes it difficult to compare the performance of the fine-tuned models and to assess their accuracy and reliability. As described in the BLEU theory section, a normal BLEU score is in the range of 20-40. Both Flan-T5-base and Flan-T5-large achieved a BLEU score on the validation set in this range as seen in Table 3. This would indicate that fine-tuning has been a success.

To put the scores in perspective, the scores can be compared to the current state-of-the-art BLEU scores on famous machine translation datasets. One popular benchmark is WMT2014 English-German [25]. The model which currently has achieved the best BLEU score according to an overview on Papers With Code is ”Noisy back-translation” from the paper ”Understanding back-translation at scale” by Edunov, Sergey, et al. [26, 27]. The paper reported a BLEU score of 35.0 for its best model which is lower than the score achieved by Flan-T5-large. However WMT2014 English-German is a much more generalized dataset containing data from multiple domains. The models getting high BLEU scores on this dataset are most likely better at translating different types of text.

5.2. Model size effects

By looking at the model sizes in Table 2 and the BLEU scores achieved on the validation set in Table 3, it is clear that increasing the number of parameters has a clear effect on the performance. The largest model Flan-T5-large with 780 million parameters achieves a 136% increase in BLEU score compared to Flan-T5-small with only 80 million parameters. This performance increase does come with its challenges. It took 22 times longer for it to converge compared to the small model. Assuming a rental cost of the hardware described in section 3.4 of $1.5 per hour, this means that the large model costs $33 to fine-tune compared to $1.5 for the smallest model

5.3. Example outputs

Table 4 in Appendix. A shows the outputs from the different models on three self-generated prompts. These prompts were made to manually assess the behavior and quality of the models.

5.3.1. Flan-T5-small

The BLEU score of 15.5 for the smallest model (as seen in Table 3) indicates that it is the weakest of the three models. Despite this, it does appear to accurately capture the meaning of the original sentences, aside from the first example. The best example is the last one, ”translate English to Norwegian: Jesus drank his wine”. In this case, the only incorrectly translated word is ”drog”, which should have been ”drakk”.

5.3.2. Flan-T5-base

The Flan-T5-base model performed surprisingly worse than expected, given its 13.5 point higher BLEU score compared to the smallest model. It did fare better on the first example, but overall, its translations were not as precise as expected from the BLEU score.

5.3.3. Flan-T5-large

The largest model produced the highest quality outputs among the three, as expected. It accurately captured the meaning of all three inputs and generated outputs that closely resembled the expected results. The only mistake made was in the first example, where it did not generate a sentence in the past tense, using ”blå” instead of ”blått”.

6. CONCLUSION

This report explores the possibilities of using pre-trained multilingual T5 models for the translation of the English to Norwegian with the goal of automatically translating english summarization datasets. This is a field of research which have had tremendous growth in the last 5 years due to the attention mechanism and transformer model. Despite there being a lot of research on machine translation and commercial options, there are no papers or benchmarks on the task of translating English to Norwegian.

The success of the models suggests that translating datasets is a realistic generation for automatically generating datasets for our master thesis. There is still a lot of work to be done to be able to do this, but this paper provides an overview of the state-of-the-art techniques for translation and valuable experience on the topic for me. Some of the further work that was out of the scope of this paper as well as possible future directions are discussed in 6.1

6.1. Further work

A more thorough review of the model's performance would provide a better understanding of its limitations and potential challenges when translating full datasets in our master's thesis. One aspect that was not covered in this project was the model's ability to handle longer inputs and maintain an understanding of word relationships. It is likely that the current models would struggle with this, and larger models such as Flan-T5-XL with 3 billion parameters or Flan-T5-XXL with 11 billion parameters may be necessary [13]. However, training these models would be computationally expensive and present its own challenges.

The Flan-T5 models used in this paper have been pre-trained on 101 different languages, including 99 that are not relevant to the task at hand. This means that they have an inherent understanding of a wide range of languages, which may be hindering their performance on the Norwegian-English translation task. A more effective solution might be to train a new set of T5 models specifically on Norwegian and English, allowing them to focus exclusively on these two languages. One way to do this could be to continue training a Norwegian T5 model using English data, such as the nb-T5-base model released by the National Library of Norway's AI Lab in 2021 [28].

One potential solution for translating the datasets may be to utilize a commercial option such as Google Translate. By accessing the Google Cloud API, we could use the same technology as Google Translate, potentially resulting in better translations than any model we train ourselves. The practical implementation and cost of this approach will depend on the size of the selected summarization dataset and is something that we will look into when writing our thesis this spring.

7. ACKNOWLEDGEMENT

I would like to acknowledge Jon-Mikkel Korsvik who I am writing my master this together with. He has been responsible for researching automatic summarization methods and I am looking forward to merging our findings into our thesis this spring. I would also like to acknowledge Habib Ullah for proofreading this paper as well as our two supervisors Kristian Hovde Lilland and Nader Aeinehchi for guidance and support.

8. REFERENCES

[1] Philipp Koehn, *Statistical machine translation*, Cambridge University Press, 2009.

[2] “A neural network for machine translation, at production scale,” `https://ai.googleblog.com/2016/09/a-neural-network-for-machine.html`, Accessed: 2022-11-28.

[3] Nal Kalchbrenner and Phil Blunsom, “Recurrent continuous translation models,” pp. 1700–1709, Oct. 2013.

[4] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le, “Sequence to sequence learning with neural networks,” *CoRR*, vol. abs/1409.3215, 2014.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” 2017.

[6] “Recent advances in google translate,” `https://ai.googleblog.com/2020/06/recent-advances-in-google-translate.html`, Accessed: 2022-11-28.

[7] “Google translate adds 10 new languages...,” `https://googleblog.blogspot.com/2008/05/google-translate-adds-10-new-languages.html`, Accessed: 2022-12-02.

[8] Taku Kudo and John Richardson, “Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing,” *arXiv preprint arXiv:1808.06226*, 2018.

[9] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur, “Recurrent neural network based language model.,” in *Interspeech*. Makuhari, 2010, vol. 2, pp. 1045–1048.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018.

[11] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al., "Improving language understanding by generative pre-training," 2018.

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[13] Raffel Colin, Noam Schick, Ashish Vaswani, Jaimie Lee, Shrimai Narang, Yonghui Liu, Lukasz Kaiser, and Illia Polosukhin, "Exploring the limits of transfer learning with a unified text-to-text transformer," *arXiv preprint arXiv:2010.10683*, 2020.

[14] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[15] Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin, "Ccmatrix: Mining billions of high-quality parallel sentences on the web," *arXiv preprint arXiv:1911.04944*, 2019.

[16] "Huggingface hub english-norwegian translation," `https://huggingface.co/datasets?language=language:en,language:no&task_categories=task_categories:translation`, Accessed: 2022-12-07.

[17] "bible-uedin," `https://opus.nlpl.eu/bible-uedin.php`, Accessed: 2022-12-05.

[18] "europa_eac_tm," `https://huggingface.co/datasets/europa_eac_tm`, Accessed: 2022-12-05.

[19] "c4," `https://www.tensorflow.org/datasets/catalog/c4`, Accessed: 2022-12-05.

[20] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel, "mT5: A massively multilingual pre-trained text-to-text transformer," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Online, June 2021, pp. 483–498, Association for Computational Linguistics.

[21] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei, "Scaling instruction-finetuned language models," 2022.

[22] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al., "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[23] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al., "Huggingface's transformers: State-of-the-art natural language processing," *arXiv preprint arXiv:1910.03771*, 2019.

[24] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," 2014.

[25] Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al., "Findings of the 2014 workshop on statistical machine translation," in *Proceedings of the ninth workshop on statistical machine translation*, 2014, pp. 12–58.

[26] "Wmt 2014 english-german," `https://paperswithcode.com/sota/machine-translation-on-wmt2014-english-german`, Accessed: 2022-12-04.

[27] Sergey Edunov, Myle Ott, Michael Auli, and David Grangier, "Understanding back-translation at scale," *arXiv preprint arXiv:1808.09381*, 2018.

[28] Nasjonalbibliotekets AI Lab, "nb-t5-base," `https://huggingface.co/NbAiLab/nb-t5-base`, Accessed: 2022-12-07.

A. APPENDIX

Prompt	Flan-T5-small	Flan-T5-base	Flan-T5-large
translate English to Norwegian: The ocean was blue	Og engel var blutt frem	Og hav var blst,	Havet var blå,
translate English to Norwegian: How are you doing?	Hvorledes er du gjort?	Hvorledes gjr du?	Hvorledes er du?
translate English to Norwegian: Jesus drank his wine	Jesus drog hans vin	Jesus drikk sin vin,	Jesus drakk sin vin.

Table 4: Selected outputs on the fine-tuned models on self-generated prompts.

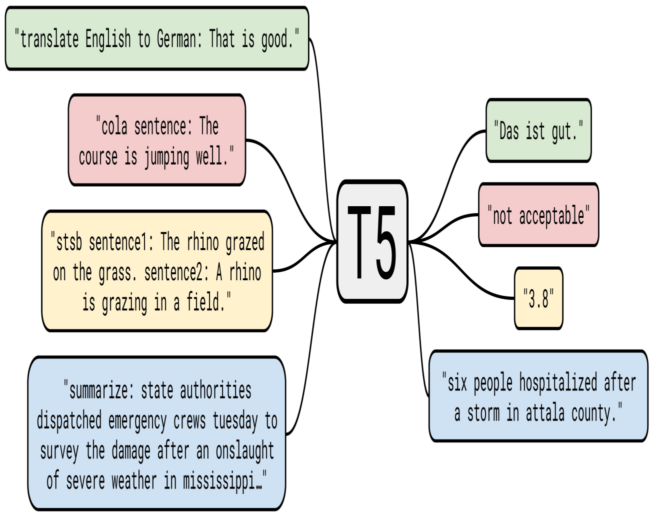
"translate English to German: That is good."
"cola sentence: The course is jumping well."
"stsb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi…"
T5
"Das ist gut."
"not acceptable"
"3.8"
"six people hospitalized after a storm in attala county."

Fig. 4: Examples of different tasks T5 has been trained on [13].