Language Translation

Abstract

"Text-to-Text Transfer Transformer" (T5) lever- aged a unified text-to-text format to get results on various NLP tasks. the pre-trained check- points for the "Text-to-Text Transfer Transformer" (T5) model released by Raffel et al. (2020) have been used to achieve state-of-the- art results on many benchmarks (Khashabi et al., 2020; Roberts et al., 2020; Kale, 2020; Izacard and Grave, 2020; Nogueira et al., 2020; Narang et al., 2020, etc.). In this paper we used mT5 amultilingual variant of T5 that was pre- trained on our multilingual dataset which was obtained from hugging face. It has 4 languages (Danish , Latvian , Chinese , Norwegian) trans- lated to English.We compared our results from the model to the results obtained from Google Translate.

Introduction

NLP makes use of transfer learning , where a model is pre-trained on a data-rich task before being fine-tuned on a downstream task of interest.(Ruder et al., 2019).Both mT5 and T5 were trained in similar fashion. The only difference was that mT5 was trained on multi-lingual data, and had vastly more token embeddings (250k). Both were initially trained on the objective of span-corruption: "consecutive spans of input tokens are replaced with a mask token and the model is trained to reconstruct the masked-out token".MT5 that treats every text processing problem as a text-to-text problem, i.e., the problem of generating some target text conditioned on the input text.Popular models of this type are mBERT (Devlin, 2018), mBART (Liu et al., 2020a), and XLM-R (Conneau et al., 2020),which are multilingual variants of BERT (Devlinet al., 2019), BART (Lewis et al., 2020b), and RoBERTa (Liu et al., 2019), respectively.

To train mT5 we used the dataset from hugging face which has all train , evaluation and test dataset.The dataset split is shown in the later section.To validate the performance of mT5 we used BLEU score to get its accuracy aganist the test dataset.

T5 is a pre-trained language model use of a unified "text-to-text" format for all text-based NLP problems. This approach is natural for generative tasks where the task format requires the model to generate text conditioned on some input.The primary advantage of this approach is that it allows the use of exactly the training objective (teacher-forced maximum likelihood) for every task, which in practice means that a single set of hyperparameters can be used for effective fine-tuning on any downstream task.

A major factor in pre-training multilingual models is how to sample data from each language.We took the first 20 percent of the dataset from each language to train the model.

The accuracy is measure by the loss function and the bleu score as a metrics. BLEU, or the Bilingual Evaluation Understudy, is a score for comparing a candidate translation of text to one or more reference translations.The BLEU score was proposed by Kishore Papineni, et al. in their 2002 paper "BLEU: a Method for Automatic Evaluation of Machine Translation".The approach works by counting matching n-grams in the candidate translation to n-grams in the reference text, where 1-gram or unigram would be each token and a bigram comparison would be each word pair. The comparison is made regardless of word order. The primary programming task for a BLEU implementor is to compare n-grams of the candidate with the n-

grams of the reference translation and count the number of matches. These matches are position-independent. The more the matches, the better the candidate translation is.

he Python Natural Language Toolkit library, or NLTK, provides an implementation of the BLEU score that you can use to evaluate your generated text against a reference.NLTK also provides a function called corpus.bleu() for calculating the BLEU score for multiple sentences such as a paragraph or a document. A perfect score is 1 which signifies a perfect translation.

Related Work

The same task is attained by various other models like mBERT , RoBERTa. mBERT (Devlin, 2018) is a multilingual version of BERT (Devlin et al., 2019). mBERT follows the BERT recipe as closely as possible (same architecture, objective, etc.) as compared to T5.The primary difference is the training set which is trained on English Wikipedia and the Toronto Books Corpus, mBERT is trained on up to 104 languages from Wikipedia.It is a multilingual encoder-decoder model that is based on BART.It is trained with a combination of span masking and sentence shuffling objectives on a subset of 25 languages.

XLM-R is an improved version of XLM based on the RoBERTa model XLM-R is trained with a cross-lingual masked language modeling objective on data in 100 languages from Common Crawl To improve the pre-training data quality, pages from Common Crawl were filtered by an n-gram language model trained on Wikipedia.

Dataset

The dataset is obtained from hugging face , the link to which is "codexgluetttexttotext".The structure of the dataset is as follows.

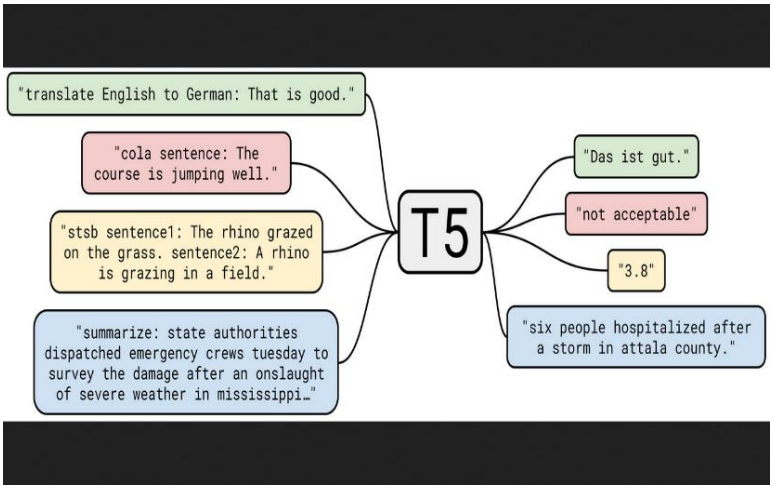
field name	type	description
id	int32	The index of the sample
source	string	The source language version of the text
target	string	The target language version of the text

Data Splits

name	train	validation	test
da_en	42701	1000	1000
lv_en	18749	1000	1000
no_en	44322	1000	1000
zh_en	50154	1000	1000

Method

T5 is a simple transformer based architecture that uses a text-to-text approach. Every task –including translation, question answering, andclassification – is cast as feeding the model text as input and training it to generate some target text. This allows for the use of the same model, loss function, hyperparameters, etc.across our diverse set of tasks. The changes compared to BERT include:adding a causal decoder to the bidirectional architecture. replacing the fill-in-the-blank close task with a mix of alternative pre-training tasks.



mT5 is similar to T5 except that it is trained on multiple languages. MT5 tokenizer converts the sentences into tokens where each token represents a word in the sentence.

Sample input text : <en> Denne artikel gennemgår et scenarie.

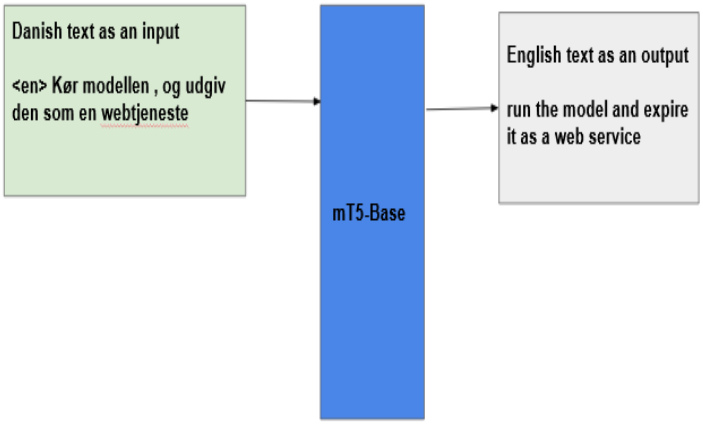
Hyperparameters

The learning rate is controlling the size of the update steps along the gradient.Choosing a good learning rate is important when training a neural network. The best learning rate is dependent on the individual problem model.Larger learning rate mean that the weights are changed more every iteration, so that they may reach their optimal value faster, but may also miss the exact optimum. Learning rate scheduling allows you to use large steps during the first few epochs, then progressively reduce the step size as the weights come closer to their optimal value. The learning rate for our model is 5e - 4 and the batch size is 4.

Optimizers in machine learning are used to tune the parameters of a neural network in order to minimize the cost function.Adaptive optimizers has been introduced to solve the issues of the gradient descent's algorithms. Their most important feature is that they don't require a tuning of the learning rate value. AdamW is good with sparse dataset and also there is no need to focus on learning rate.Optimizer.zero.grad() The function of is to set the gradient of all model parameters to 0.

The batches are chosen at random by data generator and the model gets trained.The number of epochs are 10. It would take about 5 hours on a GPU.

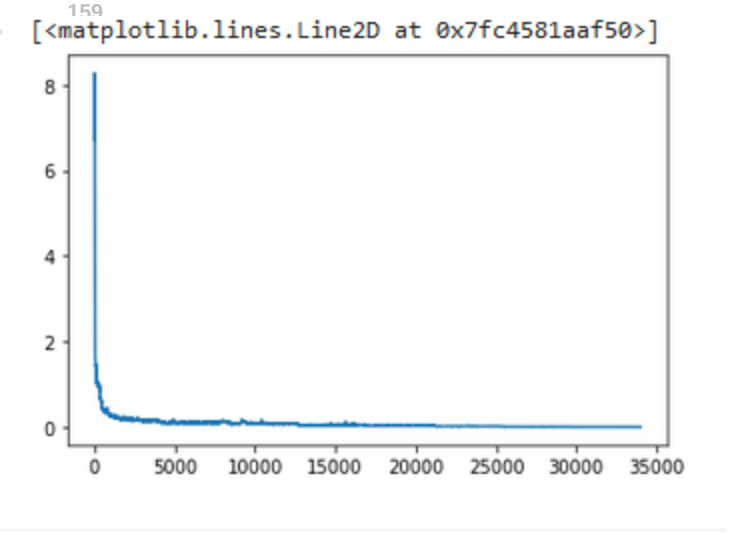
A language mapping dictionary is creates which contain the target and inout language tokens (<en> , <da> etc).Each input sentence after converted into tokens is prefixed with the token of the translator language so that the model knows the language it needs to convert to. The model is trained by choosing a random language token from this mapping.



The model is validated on the evaluation set from the original dataset and the batches are obtained using a data generator.

Results

The current results are for the translation from danish to English. It can be extended to the other languages.The model coded was hard- coded to danish to obtain the results as the running time for the entire dataset requires a TPU, limited resources and time contraints. The loss function is as follows of the model.



The bleu score we obtained was 0.5. The model is able to translate almost half of the sentences.

Conclusion and Future work

The model currently woks as expected when tested with a single language (Danish) to english. For future work we would like to train on a larger dataset amd improve the bleu score.

References

Mikel Artetxe, Sebastian Ruder, and Dani Yogatama. 2020. On the cross-lingual transferability of monolingual representations. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 4623–4637, Online. Association for Computational Linguistics.

Rodrigo Nogueira, Zhiying Jiang, Ronak Pradeep, and Jimmy Lin. 2020. Document ranking with a pretrained sequence-to-sequence model. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 708–718, Online. Association for Computational Linguistics

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In Proceedings of the 2020 Conference on Empirical Methods in Natural

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 66–75, Melbourne, Australia. Association for Computational Linguistics

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3687– 3692, Hong Kong, China. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019. Massively multilingual sentence embeddings for zeroshot cross-lingual transfer and beyond. Transactions of the Association for Computational Linguistics, 7(0):597–610.

Zewen Chi, Li Dong, Furu Wei, Nan Yang, Saksham Singhal, Wenhui Wang, Xia Song, Xian-Ling Mao, Heyan Huang, and Ming Zhou. 2021a. InfoXLM: An information-theoretic framework for cross-lingual language model pre-training. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 3576–3588, Online. Association for Computational Linguistics.

Junjie Hu, Melvin Johnson, Orhan Firat, Aditya Siddhant, and Graham Neubig. 2020a. Explicit alignment objectives for multilingual bidirectional encoders. arXiv preprint arXiv:2010.07972

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers

for language understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association

Diedre Carmo, Marcos Piau, Israel Campiotti, Rodrigo Nogueira, and Roberto Lotufo. 2020. PTT5: Pretraining and validating the t5 model on brazilian portuguese data. arXiv preprint arXiv:2008.09144.

David Crystal. 2008. Two thousand million? English today, 24(1):3–6.

Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020a. Pre-training via paraphrasing. arXiv preprint arXiv:2006.15020.