GENERATION OF COHERENT TEXT FROM NOISY DANISH SPEECH TRANSCRIPTS - PROJECT 14

USING MULTILEXNORM2021 A TRANSFORMER-BASED ENCODER-DECODER MODEL 02456 DEEP LEARNING

Oliver Wiik Rasmussen (s152920)[1, 2] and Morten G. Vorborg (s134001)[1, 2]

[1] *Stud. MSc Biomedical Engineering*
[2] *Technical University of Denmark*

(Dated: 04.01.2022)

The generation of coherent text from noisy speech transcripts results in easier readable text with fewer misunderstandings during the processing of speech to text, which saves a lot of resources. In this project, an approach to minimising the level of noise in such transcripts are presented. By fine-tuning an existing transformer-based encoder-decoder model on a dataset consisting of $1.73 \cdot 10^6$ danish words, it was possible to bring the WER, BLEU, and GLEU scores from [83.5%, 64.9%, 71.1%] to [89.9%, 75.1%, 80.7%] respectively. It is also very likely that enough information has been gathered to state that a better performing system can be developed, either by further fine-tuning the current model or by fine-tuning a larger version of the model.

I. INTRODUCTION

Generating transcriptions of speech to text is a commonly needed functionality, whether for interviews, meetings, social media texts etc. A fully working transcription model would drastically reduce the time it takes and, therefore, the money it costs. Martin Carsten Nielsen and Rasmus Arpe Fogh Jensen from Danspeech have made a speech to text transcription model to remedy the costs of doing so manually. DanSpeech has developed open-source general speech recognition models since 2018 to further the speech to text technology[1]. The output of their model is not without errors, which this paper seeks to support. Martin and Rasmus's model generally has a correct word to word ratio of 83.5% where the errors are the incorrect spelling of names, compound words being one or more words and how larger numbers are written in text.

MultiLexNorm2021 is a state of the art multilingual lexical normalisation transformer that currently is outperforming other natural language processing tools[2]. It is based on the ByT5 model [3] which is a transformer-based encoder-decoder byte-level system. MultiLexNorm2021 was made primarily due to the exploding increase in social media use which has developed slang, abbreviations and introduced many typos in its texts. It proceeded to win the W-NUT shared task in developing a system that performs lexical normalisation: the conversion of non-canonical texts to their canonical equivalent form. In particular, this task includes data from 12 languages[4].

This paper presents a further step in the transcription process that DanSpeech already performs that aims at correcting the errors the transcription model generates. To do this, the lexical normalisation transformer, MultiLexNorm2021, is fine-tuned to be specialised in the Danish language instead of a broad 12-language model; This is believed to be a possibility because the Danish data used in the original MultiLexNorm2021 was a little dated, from the social media platforms Arto and Twitter, and the number of words was limited.

II. METHODS

This section describes the data and its structure used to train the model and the approach of training the model most efficiently. The High-Performance Computing (HPC) nodes at DTUs disposal have been used to handle the extensive computer power requirements, dedicated memory and total memory.

The project has been written in python using PyTorch to fine-tune, and all of the code can be seen on the GitHub[5] which also includes a Jupyter notebook that can fine-tune a model as performed in this study, provided the data has the correct structure. As the model is too big to be in the GitHub repository, the model fine-tuned in this paper is unavailable, but one can fine-tune a new one from scratch.

A. Structure of the data

The data used in this study is from audiobooks ($\approx 1.5 \cdot 10^6$ words) and NST Danish ASR Database[6] ($\approx 0.23 \cdot 10^6$ words). Compared to the MultiLexNorm study of ca. 16.500 danish annotated words, the potential of fine-tuning a similar model is assumed to yield better results. All of the data has been annotated and structured in .txt files where every new line contains the speech recognised word followed by a tabular and then the annotated word for comparison. An empty line corresponds to the end of the sentence, followed by the beginning of the next sentence.

B. MultiLexNorm2021, a transformer-based NLP model

The MultiLexNorm2021 is a multilingual transformer-based encoder/decoder NLP model that takes a sentence as input and outputs one normalised word for that sentence. The model is based on Google Research's model ByT5. [2] The five different models of ByT5 can be seen in figure I. The chosen model to be used is the smallest one as it is still a very large model compared to general models and the computational requirements are great for a normal computer.

Size	Params	Enc	Dec
Small	300M	12	4
Base	582M	18	6
Large	1.23B	36	12
XL	3.74B	36	12
XXL	12.9B	36	12

TABLE I: List of the sizes of the possible ByT5 models

In figure 1, we see a representation of the model input, which consist of a sentence to normalise. The model normalises one word for each pass. Therefore, the sentence is represented once for each word to normalise each word individually. Sentinel tokens ($< x >$ and $< y >$) surround the word to normalise.

The MultiLexNorm2021 contains different components, which can be seen in figure 2. As aforementioned, the *Input sentence* contains sentinel tokens surrounding the word to normalise. The first part of the transformer model is the *Positional encoding* block, where the word to word relative positioning of the Input sentence is encoded. The *Encoder stack* consists of 12 *Encoder blocks* that map out the positional encoded input sentence to higher dimensionality, which is passed to each *Decoder block* in the *Decoder stack*. The output of the *Decoder stack* is an abstract numerical represented word, which is translated to a natural language word in the *Language model*. The *Language model* consists of a fully connected layer.[2, 7]

C. Training the transformer

1. Choosing the optimal fine-tuning approach

Given this project's limited time, a decision regarding the training approach has been made. Three small fine-tuning sessions have been made where most, some, or none of the blocks in the transformer have been frozen, meaning they will not be fine-tuned. Each of these sessions ran for two hours of training on around 4000 words. Two things were expected: firstly, to decide on what should be fine-tuned for longer sessions and worked with. Moreover, secondly, if the test results had better error rates than the baseline of not modulating the data at all. The error metrics are described in section II D. The results can be seen in table II.

D. Error metrics

Three different error metrics have been chosen, which will be described in this section, to test the model's performance. The reason for selecting multiple performance measurements is that they all have their qualities. It has to be mentioned that the metrics can not directly be compared to each other, but their individual changes reflect the changes in the model's performance.

1. Word Error Rate (WER)

The Word Error Rate (WER) is a performance measure based on an automatic speech recogniser's word to word accuracy. It compares a transcribed sentence called the hypothesis to the correct sentence called the reference. It is based on equation 2.1, where S is the number of substituted words, D is the number of deleted words, I is the number of inserted words, and N is the number of words in the reference. [8]

$$WER = \frac{S + D + I}{N} \tag{2.1}$$

One disadvantage of the WER is that the relation between the words in a sentence is not considered.

For easier comparison between the error metrics, the WER is chosen to be represented as the Word Accuracy (WAcc), which is defined as in equation 2.2.

$$WAcc = 1 - WER \tag{2.2}$$

2. Bilingual Evaluation Understudy (BLEU)

The Bilingual Evaluation Understudy (BLEU) is a more complex performance measurement compared to WER. The main difference between WER and BLEU is that BLEU uses n-grams to compare the relation between multiple adjacent words in the reference and hypothesis sentences. The "n" in n-grams defines the number of adjacent words to compare against each other. So, for example, with $n = 2$, in the reference sentence:

"*Social people are great*"

We compare "*Social people*", "*people are*", and "*are great*" with the hypothesis sentence. Multiple values of n are used in BLEU, and a weighted sum of the comparisons are then applied for the further calculations of the BLEU score.

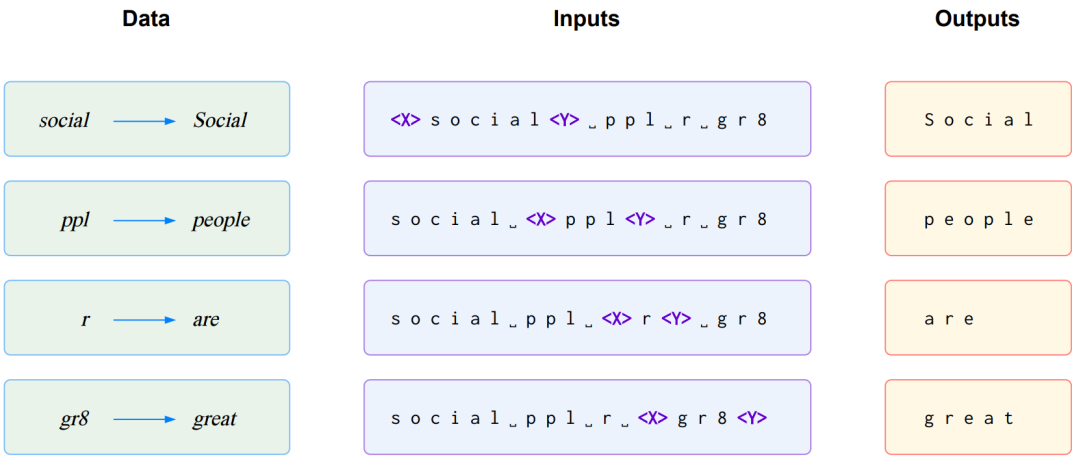

FIG. 1: The inputs and outputs of MultilexNorm2021 including the ByT5 sentinel tokens ($< x >$ and $< y >$) marking the words to be normalised. [2]

In calculating the BLEU score, both situations of over and under "guessing" are handled as well. By over "guessing", it is meant that, for example, if the hypothesis consists of multiple occurrences of only a likely occurred word like "the", one could think that a too high evaluation would be made. Instead of this overvaluation, a comparison to the maximum number of occurrences of the word is made. A penalty is given for the additional occurrences of the word. Likewise, for under "guessing", when a hypothesis is shorter than the reference, a penalty is given as well. [9]

A disadvantage concerning the BLEU is that it originally was developed for measuring text corpora; one could assume that a misvaluation could occur.[10]

3. *Google-BLEU (GLEU)*

To accommodate the possible bias of BLEU, mentioned as the disadvantage in section x Google-BLEU (GLEU) has been used as a measure as well. GLEU is also based on n-grams but is explicitly designed for sentences. [10]

A disadvantage that all the metrics have in common is that none of them considers that a word could be almost correct. For example, if the error only occurs in one letter, the entire word is considered as an error, which in many cases are not what a human would evaluate it as.

III. RESULTS

The results section provides the description and outcome after applying the defined methods in section II.

A. Fine-tuning

1. *Choosing the optimal fine-tuning approach*

Following the approach described in II C 1 table II is generated.

State of model	WAcc [%]	BLEU [%]	GLEU [%]
No modulation	83.5	64.9	71.1
Fine-tuned last decoder	83.5	64.9	71.2
Fine-tuned all decoders	85.3	65.2	73.3
Fine-tuned entire model	**86.7**	**66.2**	**74.7**

TABLE II: The error metrics output from the mini-trial decide which approach has been taken further. "No modulation" is the baseline as it describes the initial error metrics of the data before applying any model.

The outcome of the mini test resulted in that the fine-tuning of the entire MultiLexNorm2021-model was the best approach, as the output of the error metrics were somewhat higher across the board.

2. *Fine-tuned transformer performance*

After the right fine-tuning approach was decided, the full fine-tuning was started on one of DTUs HPC nodes. The training and validation loss over batch numbers can be seen in the upper figure, and the three error metrics over batch number can be seen in the lower figure of Figure 3. The model fine-tuned on $1.75 * 10^5$batches $* 5\frac{words}{batch} \approx 8.75*10^5$words which corresponds to about half of the annotated data provided before the server crashed (this will be further touched on in section IV).
The validation loss appears to reach its minimum pretty quickly, whereas the training loss keeps a downward trend

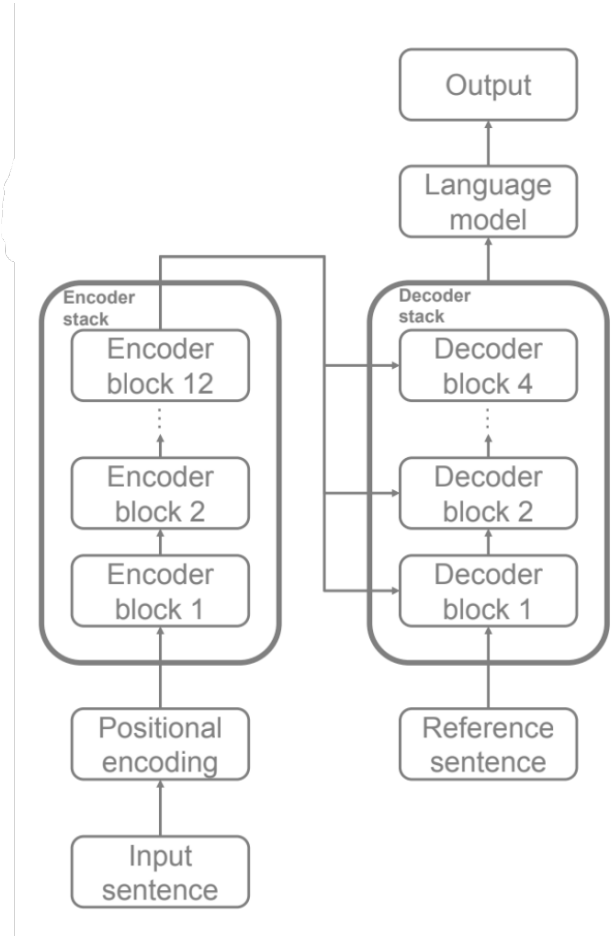

FIG. 2: Simplified model structure of the MultiLexNorm2021.

as expected. Looking at the error metric, even though the slope decreases, it does maintain a positive trend. This tells us that the model gets better over time even though it is hard to see purely looking at the loss.

The maximum values of the WAcc, BLEU and GLEU of the fine-tuned model is recorded together with the baseline of no modulation and the initial fine-tune test in table III. There is no doubt that the fine-tuning of the MultiLexNorm2021 provides a better result and fewer errors in the transcription of Danish speech to text.

State of model	WAcc [%]	BLEU [%]	GLEU [%]
No modulation	83.5	64.9	71.1
Initial test	86.7	66.2	74.7
Full fine-tune	**89.9**	**75.1**	**80.7**

TABLE III: Table containing the error metric given initial output from speech to text transcription model, the initial test described in II C 1 and lastly the result of the best performing model fine-tuning all parameters of the MultiLexNorm2021 model.

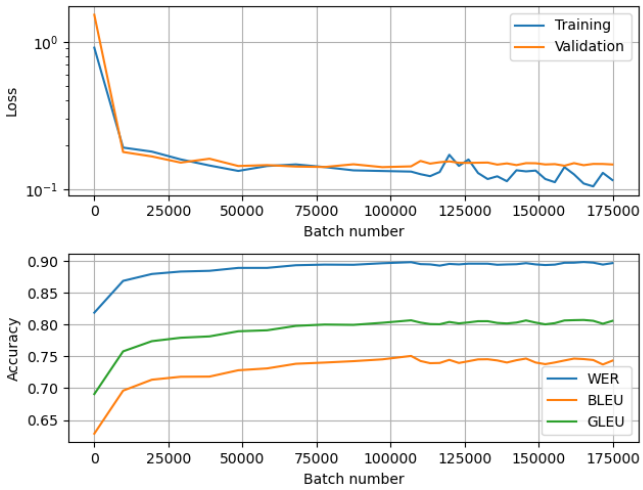

FIG. 3: Plots of the performance of the fine-tuned model. The upper plot shows the training loss in blue and the validation loss in orange. The lower plot shows the three error metrics described in section II D.

3. Examples of nomalized sentences

IV. DISCUSSION

During the workings of this paper, many obstacles have occurred regarding the fine-tuning of such a large model as the MultiLexNorm2021. Loading in a model containing $300 \cdot 10^6$ parameters requires much memory even though it is the smallest model of the MultiLexNorm2021 family. All of the provided annotated data could not be fine-tuned before the HPC nodes crashed due to lack of memory, causing the result to be incomplete. It was accommodated with saving models continuously to resume the fine-tuning if the crash happened, but that led to different patterns in the performance. This behaviour can be seen in figure 3 between 100000 and 125000 batches when the pattern changes in the error metrics plot. It is not certain if the performance change is due to overfitting or wrongly continuation of the fine-tuning. Further investigations should be made on this behaviour or could be avoided if even more memory could be available during fine-tuning. Furthermore, if the bigger models from MultiLexNorm is to be used in this project, more memory is needed no matter what as the number of parameters increases immensely.

To precisely define errors can be difficult in some of the encountered scenarios. Many of the errors regarding names was either due to the normally correct spelling of a name being wrong. Like the transcribed name Sarah being wrong because the "correct" name was Zarah or Sara, which would be impossible to correct if no larger context is present where the model should know it beforehand, or if the initial transcription model

incorrectly picked up the name.

Since the "mini-tests" conducted in section III A 1 was done for a shorter period and few words, a complete analysis should be done, fine-tuning all the examples for longer to make sure it turned out to be the better solution. As all the error metrics are so close to each other, it is not certain the correct approach was taken. However, given the time constraint, it is the best assurance there could be established at the time being.

One thing that should be re-mentioned is that almost all the annotated data used is from audiobooks, which means that the sentences used as input for the speech to text model are "perfect" in that it is not normal spoken speech. When humans talk, we often stop mid-sentence, repeat some words, say "uhm" a lot and so on. So the result might be different if the fine-tuned model is used on "real" sentences recorded directly from a human conversation and not written material read out loud.

V. CONCLUSION

This paper suggests an addition and improvement to Martin Carsten Nielsen's and Rasmus Arpe Fogh Jensen's from Danspeech speech to text model to generate more coherent text. These improvements are achieved by fine-tuning the transformer-based encoder-decoder model MultiLexNorm2021 on Danish sentences from audiobooks and the NST Danish ASR Database.

Furthermore, it is suggested that additional fine-tuning of the same size model or the same approach is taken with larger models of the MultiLexNorm as these perform significantly better in their studies but require immense time and memory to fine-tune.

ACKNOWLEDGMENTS

We want to thank Martin Carsten Nielsen and Rasmus Arpe Fogh Jensen from Danspeech for providing the challenge to try and improve on the output of their model. We would also thank for the weekly meetings which were engaging and motivating. They also brought the data and the tokenizer we used to prepare sentences for input when fine-tuning the model.

REFERENCES

[1] M. Nielsen and R. Jensen, "DanSpeech," 1 2022. [Online]. Available: https://sprogteknologi.dk/dataset/danspeech

[2] D. Samuel and M. Straka, "ÚFAL at MultiLexNorm 2021: Improving Multilingual Lexical Normalization by Fine-tuning ByT5," Tech. Rep., 2021. [Online]. Available: https://huggingface.co/ufal.

[3] L. Xue, A. Barua, N. Constant, R. Al-Rfou, S. Narang, M. Kale, A. Roberts, and C. Raffel, "ByT5: Towards a token-free future with pre-trained byte-to-byte models," 5 2021. [Online]. Available: http://arxiv.org/abs/2105.13626

[4] MultiLexNormGroup, "W-NUT," 1 2022. [Online]. Available: https://noisy-text.github.io/2021/multi-lexnorm.html

[5] O. Rasmussen and M. Vorborg, "GitHub Repository," 1 2022. [Online]. Available: https://github.com/oliverWiik/deepLearningProject.git

[6] Digitaliseringsstyrelsen, "NST Danish ASR Database," 2003. [Online]. Available: https://sprogteknologi.dk/dataset/nst-acoustic-database-for-danish-16-khz

[7] Medium, "What is a transformer," 2019. [Online]. Available: https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04

[8] M. Thoma, "Word Error Rate Calculation," 2013. [Online]. Available: https://martin-thoma.com/word-error-rate-calculation/

[9] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," Tech. Rep., 2002.

[10] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, [U+FFFD] Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation," pp. 1–23, 2016. [Online]. Available: http://arxiv.org/abs/1609.08144