Speech-to-Text for Norwegian Using Image Analysis of Spectrograms

Zoia Butenko, Martins Gintovts, Taras Andrushko
zoiab@uio.no, martigin@uio.no, tarasan@uio.no

Abstract

It is common practice in Speech Recognition to process the audio signal directly and translate it into speech. However, the state-of-the-art model Whisper released by OpenAI is based on processing images of spectrograms - a visual representation of the signal strength over time - with a transformer-based model. While being a multi-language model, Whisper doesn't perform equally well for all languages. In this paper, we investigate whether such an approach works well for the recognition of Norwegian speech commands. Given that a spectrogram can be represented in different ways, we also investigate how different pre-processing methods affect the performance of the model. We find that state-of-the-art transformer-based architectures are too sensitive to amounts of data, which could hinder progress for minority languages or even Norwegian which lacks annotated speech corpora as of now. We also find that leveraging pre-trained models trained on different data does not improve the model's performance for our specific use case.

1 Introduction

Speech Recognition is a complex problem that can be approached in a multitude of ways. On one hand, any utterance can be broken down into phonemes which can then be classified, however, it is nearly impossible to separate the sounds in a word or phrase as they usually form a continuous sequence. On the other hand, viewing speech recognition as a Sequence-to-Sequence task poses a set of challenges as well, as the number of sounds in a word rarely matches the number of symbols we use to write it down. Considering dialectical and individual variety in spoken language, speech recognition might seem almost unapproachable. Nonetheless, current state-of-the-art models demonstrate impressive results with fairly simple architectures: specifically the leverage of Visual Transformers (ViTs) [1] and Convolutional Neural Networks (CNNs) [2] as encoders seem to dominate the field. Surprisingly, best-performing models do not even use the audio signals as input: instead, image analysis is performed on spectrograms of the input audio sequence, essentially framing the task as Image Captioning. Given that visual transformers are very sensitive to the amount of training data, we hypothesize that this direction of the field's evolution is part of the problem of bias against minority languages in Language Technology. We attempt to train a Transformer-based model on a small open-source data of spoken Norwegian, partly replicating the famous OpenAI's Whisper model [3], to investigate whether current state-of-the-art is accessible for languages with fewer native speakers. Based on our results, we speculate that even for a relatively narrow domain of images such as spectrograms, the Transformer-based model requires a substantial amount of training data to be able to extract any useful features. Given this outcome, we additionally try to leverage a pre-trained ViT trained on ImageNet [4] as done by the authors of [5]. More specifically, the authors use the pre-trained model to further fine-tune it for the task of classifying spectrograms. While this approach works for the task of classification, it does not prove to be that fruitful for our specific use case.

2 Related Work

Numerous works that leverage the usage of spectrograms as inputs have been conducted. In [5], the authors introduced the Audio Spectrogram Transformer (AST), the first convolution-free, purely attention-based model, achieving state-of-the-art results for audio classification. AST takes spectrograms as inputs and outputs a label. The authors of [3] proposed the Whisper model, a multilingual model trained on 680,000 hours of transcripts of audio in the form of Log-Mel spectrograms. Currently, the model is considered state-of-the-art. [6] presents a method for speech emotion recognition using spectrograms and deep convolutional neural networks (CNN). Furthermore, the authors also investigated the effectiveness of transfer learning for emotion recognition using a pre-trained AlexNet model. [7] proposed a speech emotion recognition method based on phoneme sequence and spectrograms. The authors performed various experiments with different kinds of deep neural networks with phoneme and spectrogram as inputs and showed that a phoneme and spectrogram combined CNN model proved to be most accurate in recognizing emotions on IEMOCAP data.

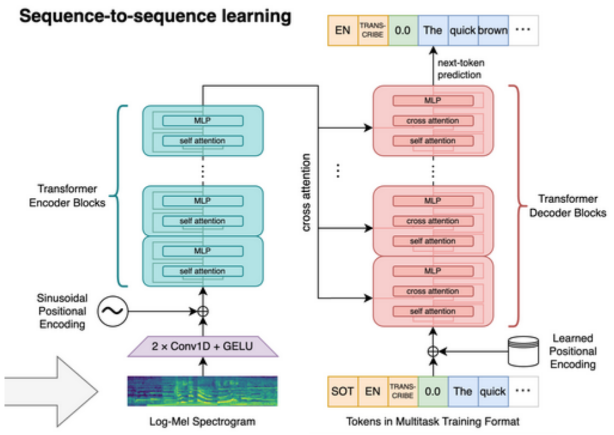
Sequence-to-sequence learning
EN
TRANS-CRIBE
0.0
The
quick
brown
next-token prediction
MLP
self attention
cross attention
Transformer Encoder Blocks
Transformer Decoder Blocks
Sinusoidal Positional Encoding
2 × Conv1D + GELU
Log-Mel Spectrogram
Learned Positional Encoding
SOT
Tokens in Multitask Training Format

Figure 1: OpenAI's Whisper architecture

3 Method

The methodology of our work can be roughly described by following steps:

1. Data pre-processing
2. Training / fine-tuning the models
3. Analyzing the results

3.1 Data pre-processing

3.1.1 Datasets and resources

The dataset is an important part of any such work, and this one is not an exception. To complete the task, the "Norwegian Voice Control Corpus"[8] dataset was chosen. The NVCC is a collection of written commands in Norwegian Bokmål and Nynorsk, along with corresponding voice recordings. These commands are similar to what you'd use on a mobile phone to perform various functions, and they represent common mobile phone tasks (e.g. 'call mom', 'set an alarm for 7 am'). The corpus consists of 9,834 queries which were recorded by 11 speakers from 5 dialect groups, each query segmented into separate audio files. All data, including transcriptions, written queries, audio segments, and speaker details, are stored in CSV files.

Unlike other Norwegian speech datasets('NB Tale – Speech Database for Norwegian'[9], 'RUNDKAST'[10], 'Norwegian Parliamentary Speech Corpus 1.1'[11]), NVCC has a better quality of recordings than others and also contains two variants of spoken Norwegian (Bokmål and Nynorsk). Moreover, there is less variety in the NVCC, and entries are much shorter on average.

3.1.2 Pre-processing

Pre-processing is one of the most crucial parts of any machine learning work; this particular one was no exception. The pre-processing plan in our case looked like this:

1. Convert .wav files into an array of digits
2. Plot the array with matplotlib
3. Apply logarithmic scale

Files were converted with the Librosa library into an array of digits using a sample rate of 22050 which is a standard of the industry. It was also necessary to decide which format our spectrograms would look like and we tested several variants, as there are a number of ways one can represent spectrograms. At first, we chose a square shape, and the resolution of each spectrogram was set to different sizes, ranging from 720x720 to 224x224. Since all audio recordings are of different lengths from approximately 1 to 16 seconds, when audio was plotted it was stretched. You can see an example of a stretched spectrogram in Figure 1.

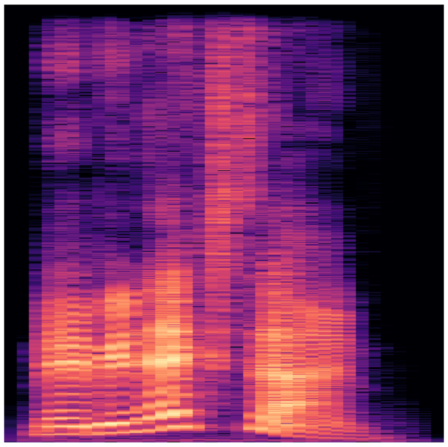

Figure 2: Example of stretched spectorgram

We can avoid this stretching if we set limits on the x and y axes. Thus, for each of the axes, a limit was chosen, for the **x-axis**, which represents time, a limit of **16 seconds** was set, since this is the longest record in the case, and for the **y-axis**, a limit of **8000** hertz was set, since everything higher is no longer a voice, but high-frequency noise. If any audio contained less than 16 seconds of information 0 were added to the end of an array.

Thus, with padding, we avoided the problem that the same sounds in different audio recordings were reflected differently and the sounds stopped stretching, so it would potentially be easier for the model to learn. An example of pre-processed audio can be seen in Figure 2 (representing the same audio recording as in Figure 1).

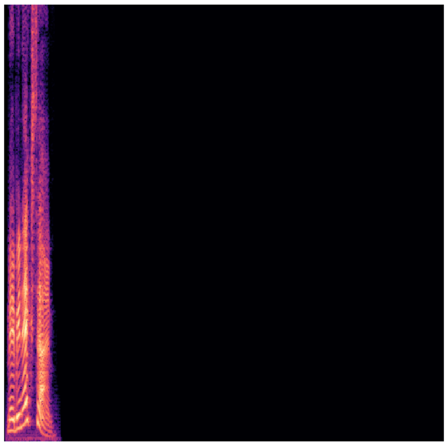

Figure 3: Example of padded audio

We also experimented with Log-Mel black and white spectrograms, where we attempted to produce rectangular shapes. Instead of plotting the image, we directly saved it by using both Librosa and scikit-image libraries. Each spectrogram had a different width, so we padded it with zeros to the longest spectrogram in the corpus directly in the dataloader. An example of a Log-Mel spectrogram can be seen in Figure 4.

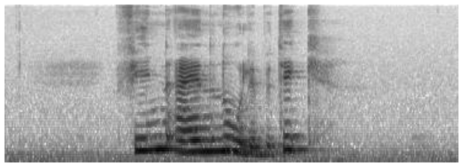

Figure 4: Example of rectangular Log-Mel spectrogram

3.2 Training / fine-tuning the models

Given that we experiment with several architectures, as well as different pre-training / fine-tuning techniques, a more detailed overview of methodology is provided in Section 4. While there are architectural differences, in most of the experiments we utilize 2 AdamW optimizers with different learning rates for the encoder and decoder, custom text tokenizer or the one derived from NorBERT model [12] - autoregressive model pre-trained on Norwegian texts. In the case of the latter, we manually add beginning- and end-of-sequence tokens. Given the nature of our data, our custom tokenizer simply splits the words on whitespace. We train all models between 2 to 5 epochs. We retain 90% of our data for training and 10% of data for evaluation. An overview of architectures used is provided in Table 1.

Table 1: Full architectures of Img2Text models

Encoder	**Decoder**	**Enc/Dec lr**	**Tokenizer**
Base Transformer	Base Transformer	1e-3; 1e-3	NorBERT
Pre-trained ViT	Base Transformer	1e-4; 1e-3	NorBERT
Pre-trained ViT	LSTM	1e-3; 3e-4	Custom
ResNet	LSTM	1e-3; 3e-4	Custom

When attempting to pre-train our own transformer decoder to be better at generating Norwegian text, we make use of a transformer decoder consisting of 4 decoder blocks and pre-train it on Norwegian

Newspaper Corpus[1]. Given our restricted computing and time resources, we limit ourselves only to a small portion of data, comprising 1M text fragments. The fragments have different lengths, varying from 4-5 word long phrases up to 800 word long paragraphs.

4 Experiments and Results

4.1 Establishing architecture

As mentioned in Section 1, being inspired by the Whisper model, for this task we are motivated to utilize a transformer architecture, namely leveraging transformers' encoder and decoder. A notable difference between our implementation and that of Whisper's lies in how input to the transformer encoder is processed. Whisper model utilizes 2 1D-convolution layers with GeLU activation functions, followed by a sinusoidal positional encoding. Our approach, as described in the next subsection leverages patches identical to that of regular ViT.

Given that images of spectrograms belong to arguably a very narrow and niche domain, our initial thinking is based on training the model from scratch. Since a great number of image models are trained on images containing natural-world objects, we do not expect the model weights learned from such data to be helpful in learning features from images of spectrograms for Norwegian speech. Hence, the idea of using a model pre-trained on e.g. ImageNet is initially abandoned by us. However, this ultimately does not result in any good performance, which we suspect is caused due to the transformers' need for a very large number of data samples to work efficiently.

4.1.1 Encoder

Investigating the problem further, we come across a study [5], where the authors show that leveraging a ViT pre-trained om ImageNet data in fact proves to be useful for classifying spectrograms, despite the considerable domain differences between the two datasets. Inspired by it, we proceed to utilize the same pre-trained model that the authors of the study did - a data-efficient image Transformer (DeiT) [13], which is trained with CNN knowledge distillation, 384×384 images, has 87M parameters, and achieves 85.2% top-1 accuracy on ImageNet 2012.

As reported by the authors, given that spectrograms tend to have a rectangular shape and thus are shorter in height dimension and longer in width dimension, their shape differs from the fixed input shape of ViT. Indeed, as described in Section 3.1.1, the shape of spectrograms we utilize and the input shape of ViT are not identical. This essentially means that (1) there will be many more patch embeddings for the height dimension, and (2) there won't be enough learned patch embeddings for the width dimension. In other words, there will be a shape mismatch. To overcome this issue, we replicate the methodology of the authors: we make use of cut and bi-linearly interpolate methods for positional embedding adaptation. As an example, given that ViT takes 384×384 image input and uses a patch size of 16×16, the number of patches in each dimension is equal to 24. Let us say that the spectrogram image has a shape of 128 x 1040. The number of patches is then equal to 8 in height dimension and 65 in width dimension. Moreover, given the nature of spectrograms, we want to leverage the use of patches with overlap (in our work, we utilize an overlap of 6 in both dimensions), so as to steer clear of potentially splitting a unit sound into several chunks. This results in an even greater number of patches. We want to avoid randomly initializing and learning extra patches from scratch, as this leads to poorer performance. We therefore (1) cut the first dimension by cutting from the middle and extracting "edge" patches, and (2) interpolate the second dimension of the 24×24 ViT positional embedding to spectrogram patches. Finally, we also remove the positional embeddings for distillation and classification tokens, as these are not required for our specific task.

Finally, despite the fact that we wish our model to be based on a Transformer encoder, after achieving unsatisfying results with the proposed architecture, we proceed to test whether a simpler

[1]https://www.nb.no/sprakbanken/en/resource-catalogue/oai-nb-no-sbr-4/

CNN encoder such as ResNet [14] would potentially serve better as an encoder.

4.1.2 Decoder

After achieving low performance using a transformer decoder and training it from scratch on our tiny dataset, as mentioned in Section 3.2, we make an attempt to first pre-train a decoder on greater amounts of Norwegian text, before further fine-tuning it for our use case. While such a decoder produces a slightly more coherent text, it is still far from being sufficient. Consequently, given our small amount of data, we speculate that it is potentially wise to instead roll back and downgrade our decoder to leverage a simpler architecture, such as LSTM architecture with attention. As expected, after substitution the text produced by the model becomes far more comprehensible and accurate. More specifically, we use a single LSTM layer employing Bahdanau Attention [15]. We initialize our hidden state by taking the mean of the encoder output - that is, taking the mean of all patch embeddings produced after encoding the spectrogram.

4.1.3 Results of using different architectures

As pointed out in the previous sections, given our small dataset of spectrograms of Norwegian speech commands, none of the proposed architectures produce any fruitful results. While the LSTM decoder manages to generate quite coherent text and captures the overall structure of the language, during the evaluation the whole model proceeds to output the same phrase for almost every input given. In this regard, a model utilizing the ResNet-encoder exhibits the most interesting results: while occasionally generating identical text given different spectrograms, its outputs vary from input to input. Examples of target captions and captions generated by ResNet + LSTM can be observed in Table 2. A translation from top to bottom with the generated caption and its corresponding target sequence separated by a dash is as follows: "*I don't want to have subtitles in the movie*" - "*look up romance movies please*" and "*I want you to open the email I received 22. march*" - "*please check norli opening times*". As we can see, the content of the captions doesn't correlate.

Moreover, as depicted in 5 during the training of any of the proposed architecture, the training loss starts to decrease but proceeds to increase after hitting the 3rd or 4th epoch. We suspect that at first, a completely incoherent text becomes more structured and grammatical which leads to a decrease in the loss, followed by the model's failure to relate the encoded spectrogram to its caption, which in turn leads to an increase in the loss. Given the fact that each model behaves in a similar fashion, we hypothesize that the encoder is the bottleneck here. It is oftentimes the truth in Image Captioning tasks - an encoder-decoder model is as good as the encoder. It is also worth mentioning that it is difficult to tell whether the way the spectrograms were processed had any big impact on performance since our models didn't perform well despite the difference in the pre-processing stage.

Table 2: Generated and target captions

Generated	**Target**
jeg vil ikke ha på subtitles på filmen	søk opp filmar med romantikk takk
jeg vil at du skal åpne e-posten som eg fekk toogtyvende mars	vær så vennlig å sjekke åpningstidene til norli

5 Discussion

As mentioned above, all model configurations tested have shown poor results. Although annotations generated by the decoder are correct grammatically and semantically (i.e. the content of each annotation makes sense in isolation), they do not seem to correlate with the corresponding input sequence.

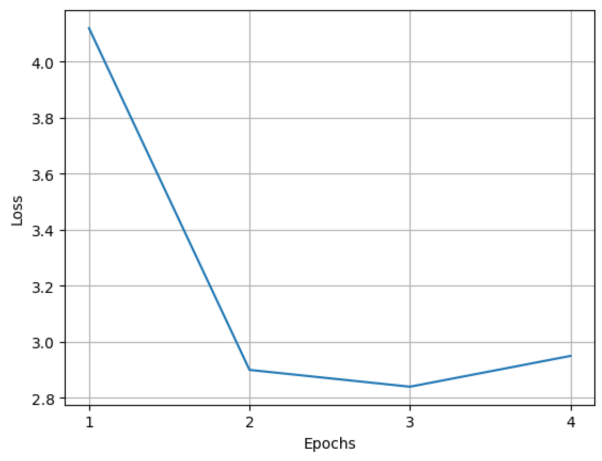

Figure 5: Loss curve of ResNet

This suggests that the decoder has trained well enough on textual data to produce coherent annotations, while the encoder could not be trained to provide valid features in order for the annotation to be possible. We therefore do not provide any metrics related to the model's performance, specifically for the following reasons:

1. Measuring the accuracy of text generation is a complicated task, with the most popular metrics – BLEU and ROUGE – also being the most heavily criticized. Matching the output of the model to the annotation and counting accuracies as 0s and 1s would not reflect the model's actual performance, leading to a lot of false negatives.

2. Any instances of the output being considered correct under evaluation could be a coincidence considering the nature of the data: short phrases with a small variety in vocabulary.

We insist that the amount of input data is the crucial factor in the model's poor performance. The nature of speech data does not allow for an easy expansion of the corpus (unlike textual data which could always be expanded by web-scraping, for example), however, it allows for various data augmentation techniques to be implemented. For example, pitch could be adjusted for gender differences (with women having higher voices than men on average), doubling the corpus. Additionally, the recording can be stretched or compressed to simulate slower and faster speech. Although the corpus used consists of very clean recordings, background noise from other datasets can be added to simulate a more natural environment as well as augment the dataset. Finally, data augmentation can be maximized by applying the features described above at each epoch (although the number of epochs used to train transformers tends to not exceed 3, which would not make such a significant difference as for architectures that require 100+ epochs at training). Considering all the aforementioned data augmentation techniques we could probably end up with a dataset of hundreds of thousands of spectrograms. However, even a dataset of such size would not guarantee decent performance considering that transformers usually require 10M+ data entries to train. Additionally, along with an increase in dataset size, computational resources, and memory consumption would increase as well. Having restricted time and resources, we did not attempt to augment data, although it could potentially prove very useful in future research.

Finally, data for other languages can be used to improve performance, especially those linguistically and phonetically similar to Norwegian, such as Danish and Swedish. One of the advantages of the Whisper model is that it is multilingual, which can greatly impact performance since the model is able to learn features even from data of very distant languages (as there are only so many sounds humans can articulate).

6 Conclusion

Two main conclusions can be drawn from this work. Firstly, pre-trained models can, as pointed out by the authors of [5], be useful for downstream classification tasks, even for very domain-specific data such as spectrograms. That, however, doesn't seem to apply to Image Captioning. Our preliminary study shows that this could be due to two reasons: (1) Image captioning being a harder task to solve than classification and (2) more data is needed for such a task. Secondly, state-of-the-art transformer-based models for speech recognition are highly sensitive to the size of the training dataset, making them mostly suitable for the most popular languages. Languages with smaller numbers of native speakers (such as Norwegian) and especially minority languages rarely have enough training data which could reinforce already existing linguistic and technological bias. Lack of data can be potentially counteracted with data augmentation techniques or using data for other languages, however, it is unclear whether open-source data would be enough even when augmented and needs further investigation.

7 Group Contributions

As with all the previous projects, our work is iterative in nature, meaning that while everybody has their set of responsibilities and work to conduct, regular meetings and discussions were ubiquitous. Therefore, it is necessary to emphasize the authors' constant involvement in each other's work. In this project, Taras Andrushko was responsible for all the pre-processing of the data, its code, corpus management, as well as theoretical research on audio-to-spectrogram conversion. Martins Gintovts found appropriate models, established and coded the architectures, in addition to searching for relevant work done for this task. Zoia Butenko came up with the idea of the project as a whole, thus being responsible for the introduction, and interpretation of the results while bringing everything together.

References

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," 2021.

[2] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015.

[3] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," 2022.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[5] Y. Gong, Y.-A. Chung, and J. Glass, "Ast: Audio spectrogram transformer," 2021.

[6] A. M. Badshah, J. Ahmad, N. Rahim, and S. W. Baik, "Speech emotion recognition from spectrograms with deep convolutional neural network," in *2017 International Conference on Platform Technology and Service (PlatCon)*, 2017, pp. 1–5.

[7] P. Yenigalla, A. Kumar, S. Tripathi, C. Singh, S. Kar, and J. Vepa, "Speech emotion recognition using spectrogram & phoneme embedding." in *Interspeech*, vol. 2018, 2018, pp. 3688–3692.

[8] "Norwegian voice control corpus," https://www.nb.no/sprakbanken/en/resource-catalogue/oai-nb-no-sbr-75/, 15.12.2022.

[9] "Nb tale – speech database for norwegian," https://shorturl.at/irwLX, 22.12.2015.

[10] I. Amdal, O. Strand, J. Almberg, and T. Svendsen, "Rundkast: an annotated norwegian broadcast news speech corpus." 01 2008.

[11] P. E. Solberg and P. Ortiz, "The norwegian parliamentary speech corpus," 2022.

[12] A. Kutuzov, J. Barnes, E. Velldal, L. Øvrelid, and S. Oepen, "Large-scale contextualised language modelling for Norwegian," in *Proceedings of the 23rd Nordic Conference on Computational Linguistics (NoDaLiDa)*, S. Dobnik and L. Øvrelid, Eds. Reykjavik, Iceland (Online): Linköping University Electronic Press, Sweden, May 31–2 Jun. 2021, pp. 30–40. [Online]. Available: https://aclanthology.org/2021.nodalida-main.4

[13] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers distillation through attention," 2021.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[15] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2016.