Training a Speech-to-Text Model for Dutch on the Corpus Gesproken Nederlands ⋆,⋆⋆

Willem Röpke, Roxana Rădulescu, Kyriakos Efthymiadis, and Ann Nowé

Artificial Intelligence Research Group, Vrije Universiteit Brussel, Belgium
{Willem.Ropke,Roxana.Radulescu,Kyriakos.Efthymiadis,Ann.Nowe}@vub.be

Abstract. Speech-to-text, also known as Speech Recognition, is a technology that is able to recognize and transcribe spoken language into text. In subsequent steps, this transcription can be used to complete a multitude of tasks, such as providing automatic subtitles or parsing voice commands. In recent years, Speech-to-Text models have dramatically improved thanks partially to advances in Deep Learning methods. Starting from the open-source project DeepSpeech, we train speech-to-text models for Dutch, using the Corpus Gesproken Nederlands (CGN). First, we contribute a pre-processing pipeline for this dataset, to make it suitable for the task at hand, obtaining a ready-to-use speech-to-text dataset for Dutch. Second, we investigate the performance of Dutch and Flemish models trained from scratch, establishing a baseline for the CGN dataset for this task. Finally, we investigate the issue of transferring speech-to-text models between related languages. In this case, we analyse how a pre-trained English model can be transferred and fine-tuned for Dutch.

Keywords: Speech Recognition · Corpus Gesproken Nederlands · DeepSpeech · Speech-to-Text

1 Introduction

The field of Artificial Intelligence has managed to achieve unprecedented results in the last couple of decades, across many fields, ranging from computer vision, e.g., beating human performance at the image recognition task [31], to reinforcement learning, e.g., achieving superhuman performance at the game of Go [33]. Many of these developments are already being applied to everyday life, in the form of product recommendation systems [32], vehicles with partially autonomous capabilities [9] or financial fraud detecting systems [36].

This work will take a closer look at another one of these fields, namely Speech-to-Text (STT), otherwise referred to as Speech Recognition (SR). Simply put, speech-to-text is the ability of a machine to "hear" spoken language and transcribe it. In subsequent steps, this transcription can be used to perform certain

⋆⋆ Part of this work was carried out by the first author for his bachelor thesis [30], under the supervision of the other authors.

actions or tasks. In recent years we have seen the practical application of this process in consumer directed systems such as Google Assistant[1], Siri[2], or Alexa[3] [21]. STT as a field has many other practical applications, ranging from healthcare [7] and helping people with disabilities [23], to controlling military aircrafts [5, 34]. This potential societal impact makes STT an attractive research subject. However, STT has faced a multitude of difficulties along its history. For starters, early models were simply not expressive enough to grasp the complexity of human speech and later models like artificial neural networks required massive datasets and substantial computing power, which up until recently have not been available.

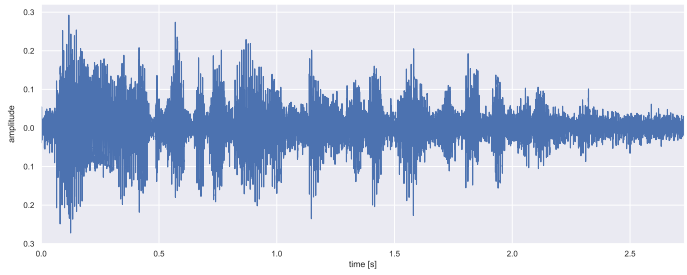
A vital component for obtaining high quality STT systems is thus having a large and well curated dataset. There are currently numerous STT datasets available for English, such as LibriSpeech [26], TED-LIUM [16] and VoxCeleb [2, 22]. However, projects such as Common Voice[4] aim to facilitate the creation of community curated datasets for a larger variety of languages. It currently contains data across 29 languages, including 23 hours of recordings for Dutch. This relatively small size has led us to look for a different dataset. For this work we have selected the Corpus Gesproken Nederlands [24, 35][5]. This corpus consists of 900 hours of spoken Dutch in both the Flemish and Dutch dialects. A detailed description together with how we pre-process CGN for this work is presented in Section 4.

For training our speech-to-text Dutch models we rely on Mozilla's open-source project DeepSpeech[6], which offers an implementation of the neural architecture presented by Hannun et al. [14]. This allows us to train speech-to-text models for Dutch, using our desired data. We offer an overview of the theoretical concepts involved in creating this deep learning architecture in Section 2, while Section 3 presents other lines of work related to this topic.

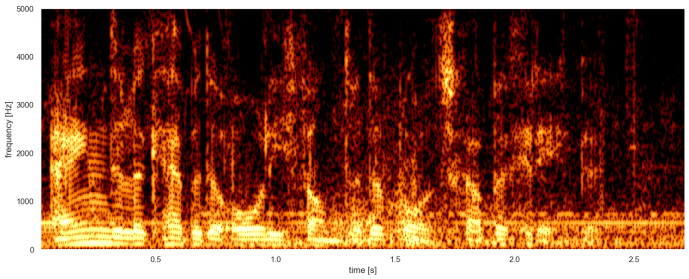
We investigate two types of models trained using DeepSpeech on the CGN. First we consider training a model from scratch and establish a baseline for how well CGN can perform when training an end-to-end speech-to-text model on a neural architecture. Afterwards, we also investigate if we can boost the obtained performance by taking advantage of a high performing model trained on an extensive English dataset and fine-tune it using CGN. An overview of our results is presented in Section 5.

Contributions We can summarize the contribution of our work as follows: (i) we provide a full analysis and contribute a pre-processing pipeline for the CGN dataset aimed for use in an end-to-end neural architecture for speech-to-text; (ii) we train and evaluate STT models for Dutch using an end-to-end deep

[1] `https://assistant.google.com`
[2] `https://www.apple.com/siri/`
[3] `https://developer.amazon.com/alexa`
[4] `https://voice.mozilla.org/en/datasets`
[5] `https://ivdnt.org/downloads/tstc-corpus-gesproken-nederlands`
[6] `https://github.com/mozilla/DeepSpeech`



(a) Waveform of the utterance "eindelijk hebben de olifanten de boodschap begrepen".



(b) Spectrogram of the utterance "eindelijk hebben de olifanten de boodschap begrepen".

Fig. 1: Two different representations of the same utterance.

learning approach in the form of the open-source project DeepSpeech, providing a benchmark for Dutch models on the CGN dataset; (iii) we investigate and evaluate the possibility of transfer learning between English and Dutch STT models.

2 Background

2.1 Speech Spectrograms

The task of speech-to-text is to find a mapping between natural language utterances and their corresponding written form representation, i.e., a textual transcriptions. The input for the DeepSpeech end-to-end deep learning systems consists of the representation of the audio file in the frequency domain, i.e., a spectrogram. Speech spectrogram allows one to visualise the spectrum of frequencies of an audio signal and identify phonetic information of the given input. Figure 1b presents an example for the spectrogram of the utterance "eindelijk hebben de olifanten de boodschap begrepen" (a fragment from a Flemish audio file in component h, lessons recorded in a classroom, Table 1), while Figure 1a

presents the same utterance as a waveform plot (representations created using the Parselmouth library [19]).

2.2 Recurrent Neural Networks

Speech data has a sequential character, meaning that order plays an important part in understanding or processing the given information. This has proven hard to tackle using regular feed-forward neural networks. Recurrent neural networks (RNNs) are artificial neural networks specialized in processing sequential data [10] (e.g., time series, DNA sequences or speech).

Given an input x representing a time-series of length T, we are interested in predicting a sequence of character probabilities that will form our transcription. After obtaining this prediction, DeepSpeech uses the CTC loss [11] to measure the prediction error for training the network. Every component of x consists of a vector of audio features, in the form of frequency information as detailed above. The characters we consider in order to train our Dutch models belong to the set: {a, b, c, ..., z, ', &, -, *space*}. We have included the special characters ', & and - because they were already present in multiple utterances in the CGN. It would be feasible to exclude these characters from the set with some extra pre-processing steps.

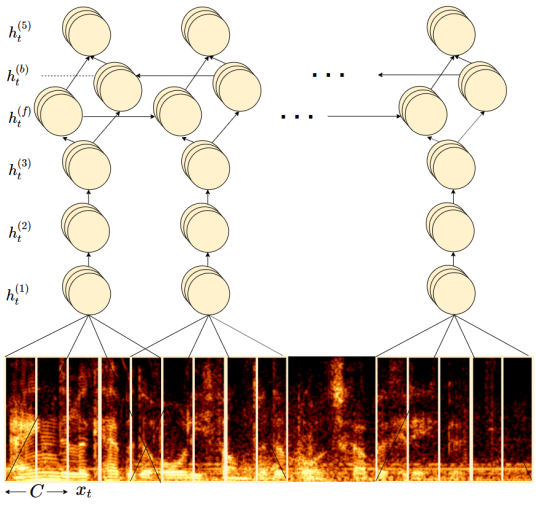


Fig. 2: Recurrent neural architecture for the DeepSpeech project, figure adapted from [14].

Figure 2 presents the architecture used in the DeepSpeech project. The network has a simple architecture, consisting in 5 hidden layers, out of which only the fourth one is a bi-directional recurrent layer with forward ($h_t^{(f)}$) and backward ($h_t^{(b)}$) units. This allows the network to not only capture past information

at a given time-step t, but to actually output predictions that depend on the entire input sequence, as we have two recurrent layers that move both forward (from the start to the end of the sequence) and backwards (from the end to the start of the sequence) through time. When passing through the first layer, for every time-step t, we consider both the spectrogram frame x_t, together with the surrounding C frames on each side.

This architecture has led to very good results when trained on English conversational speech data, reaching a word error rate of 16% on the Switchboard dataset, when trained on both the Switchboard and Fisher datasets [14]. More recent models using this architecture have scored a word error rate as low as 8.22% on the LibriSpeech clean test corpus.

Auxiliary Language Model Because an RNN will output a character by character transcription of the input, the network will often produce phonetically correct interpretations, but incorrect spellings. To remedy this, an *N-gram language model* is attached to the output. What this language model does is to try and find the most likely sequence of words instead of characters. Afterwards the sequence that maximizes a scoring function that takes the output of the RNN and the language model into account is chosen as the final output.

3 Related Work

Advances in the field of Deep Learning, together with newly developed architectures have led to spectacular results for the speech-to-text domain [1, 12–14]. However, state-of-the-art models are usually trained and benchmarked on English datasets.

Research into STT using neural architectures for Dutch has been very limited, with a few exceptions [8]. However, approaches based on Hidden Markov Models (HMMs) and statistical language models has been extensively explored in the literature [3, 6, 17, 25]. A notable initiative was Project SPRAAK[7] [4, 27], an open source speech-to-text package for Dutch, also based on an HMM architecture. Kessens and Van Leeuwen [20] have also proposed the N-Best 2008 benchmark dataset (i.e., the news broadcast and telephone dialogue components from the CGN), in order to evaluate the SPRAAK system.

For our work, we rely on the architecture developed by the open-source project DeepSpeech. Other similar frameworks include PyTorch-Kaldi [29] and wav2letter++ [28]. Finally, a similar initiative to this current work has also been applied to other languages, as for example Russian [18].

4 Pipeline for Processing the Corpus Gesproken Nederlands

The Spoken Dutch Corpus is a database of Dutch as spoken by adults from the Netherlands and Flanders. Its contents entail almost 9 million words, adding to a

[7] `https://www.spraak.org`

total of 900 hours of spoken Dutch. The goal of this project was to create a corpus that would form a plausible sample of contemporary Dutch as spoken in the Netherlands and Flanders. In creating the corpus, the developers have attempted to assemble it to optimally suit the needs of diverse research disciplines and applications. For this reason, all data is sorted by type of speech into multiple components as shown in Table 1. Each component is divided into Flemish audio files and Dutch audio files. This represents a split of 76.23% and 23.77% of Dutch and Flemish audio files respectively.

Table 1: Components available in the Corpus Gesproken Nederlands dataset with their corresponding descriptions.

Component	Description
comp-a	spontaneous conversations (face-to-face)
comp-b	interviews with teachers of Dutch
comp-c	spontaneous telephone dialogues (recorded via a switchboard)
comp-d	spontaneous telephone dialogues (recorded on MD with local interface)
comp-e	simulated business negotiations
comp-f	interviews/discussions/debates (broadcast)
comp-g	(political) discussions/debates/meetings (non-broadcast)
comp-h	lessons recorded in a classroom
comp-i	live (eg sport) commentaries (broadcast)
comp-j	newsreports/reportages (broadcast)
comp-k	news (broadcast)
comp-l	commentaries/columns/reviews (broadcast)
comp-m	ceremonious speeches/sermons
comp-n	lectures/seminars
comp-o	read speech

Out of the box, the Corpus Gesproken Nederlands does not come ready to efficiently train a speech-to-text model. Neural-based STT architectures, including DeepSpeech, will not perform well when trained on long audio files. The usual approach is to split each file in smaller chunks and then use those for the training process. As a consequence, we have processed the CGN to better fit our needs. We do this in multiple parts, starting with first splitting the audio files into shorter segments, and afterwards presenting two approaches targeted at reducing the noise in the dataset. All the code used to build this pipeline is available at: `https://github.com/wilrop/Import-CGN`.

4.1 Segmenting the Audio Files

The average length of an audio file in the original dataset is approximately 234 seconds, or 3 minutes and 54 seconds with a standard deviation of 283. Having a dataset for STT with audio files this long and a variance this high would make it

very difficult to efficiently train a model. For this reason we have opted to split audio files into smaller segments.

In the XML transcriptions of the corpus, timestamps for the beginning and ending of each sentence in an audio file were provided. We used these timestamps to group several sentences together in a new audio file until the length of this file would reach a certain threshold, in our case 3 seconds. Using this strategy we had a good balance between the amount of files and the length of these files. The Flemish set now has a mean of a bit more than 6 seconds with a standard deviation of 1.6 and the Dutch set has a mean of 6 seconds with a standard deviation of 1.5.

4.2 Handling Noise in the Dataset

After analysing the output of our first iteration of models it quickly became obvious that we were dealing with an excessively noisy dataset, thereby making it more difficult to accurately train and score our models. To tackle this problem, we have developed two methods.

Our first solution to this problem was to generate a smaller dataset, that would exclude certain components completely. We hypothesized that a lot of the noise in the data comes from face-to-face speech or spontaneous dialog, a noisy settings even for humans. These components would have a greater likelihood of people interrupting each other or talking over each other, resulting in incorrect transcriptions. We have identified components a, c, d, f and g to contain the highest amount of this type of speech (see Table 1 for a description of these components). By excluding these components we hoped to still have an adequate amount of audio files in the dataset, but with better quality. This dataset will also be referred to as the *small* dataset from here on after.

The second method tries to tackle the noise in the dataset in a different way. When inspecting transcriptions in the dataset, we noticed that several times, completely different transcriptions would start at overlapping timestamps. Because there is no way to know which transcription, if any, is correct in this case without a laborious manual process, we decided to leave out all instances where this issues occurred. Using this method we have generated the *non overlapping* dataset. Finally, we have also created a third dataset, by combining the two noise handling techniques presented above: the *small non overlapping* dataset. These partitions were generated for Dutch, Flemish and also for a combination of the two, having obtained in the end 9 datasets. Tables 2, 3, 4 present the amount of hours present per split in each of the created datasets.

5 Training a Speech-To-Text Dutch Model

We present here the results we obtained for each subset we generated from the CGN dataset, for both models trained from scratch and also by fine-tuning a state-of-the-art English model.

Table 2: Amount of hours of audio in the splits for each Flemish dataset.

	Train	Validation	Test
Small non overlapping	24.2	6.0	7.6
Non overlapping	34.1	8.5	10.6
Small	60.3	15.0	18.8

Table 3: Amount of hours of audio in the splits for each Dutch dataset.

	Train	Validation	Test
Small non overlapping	39.2	9.8	12.2
Non overlapping	61.3	15.3	19.2
Small	81.8	20.5	25.6

Auxiliary Language Model for Dutch Similar to the original approach present by Hannun et al. [14], to aid our prediction process we have created an 5-gram language model for Dutch. We have done this only once, because the Dutch and Flemish dialects follow the same grammar and spelling. The 5-gram language model used in our experiments was trained on a collection of over one million phrases, adding up to a vocabulary of more than 160 thousand words, gathered from the CGN. We used KenLM[8], which is shown to provide faster and smaller language queries than other systems [15] to efficiently generate a language model.

5.1 Experimental Setting

Splitting the Dataset We have opted to split each dataset into 80%–20% for training and testing. We also create a validation dataset by making again an 80%–20% split on the training data.

In order to obtain high-performing models, we do a parameter search for the number of hidden nodes per layer, learning rate and dropout, as shown in Table 5. We fix the rest of the parameters and present the exact values used during the training process in Table 6.

The metrics used to evaluate our results are the Word Error Rate (WER) and Character Error Rate (CER). The CER is computed as the Levenshtein distance,

[8] https://github.com/kpu/kenlm

Table 4: Amount of hours of audio in the splits for each combined dataset.

	Train	Validation	Test
Small non overlapping	63.4	15.9	19.8
Non overlapping	95.4	23.8	29.8
Small	142.1	35.5	44.5

Table 5: All variables and their values experimented with as parameters during training.

Variable	**Values**
Amount of hidden nodes per layer	512, 1024 and 2048
Learning rate	0.001, 0.00055 and 0.0001
Dropout	0.2, 0.35 and 0.5

Table 6: Flags set in training of our models that were kept constant.

Flag	**Value**
–train_batch_size	64
–dev_batch_size	32
–test_batch_size	32
–epochs	30
–use_warpctc	True
–early_stop	True
–es_steps	10
–es_mean_th	0.1
–es_std_th	0.1

which represents the minimum number of operations required to transform the ground truth into the output, divided by the total amount of characters in this ground truth. The WER is defined as the Levenshtein distance on word level divided by the amount of words in the original text.

5.2 Training a Dutch Model from Scratch

We have trained models for each of the sub-datasets we created for Dutch, Flemish and the combined data: *small*, *non-overlapping* and the *small non-overlapping datasets*. This has also allowed us to evaluate the noise handling techniques developed for CGN.

Table 7: Best performing models on the Flemish datasets.

	Nodes	**Learning Rate**	**Dropout**	**WER**	**CER**
Small Non Overlapping	2048	0.00055	0.35	0.386307	0.231521
Non Overlapping	2048	0.00055	0.35	0.469610	0.295521
Small	2048	0.00010	0.35	0.519958	0.366938

Table 7 presents our results on the Flemish data, together with the parameters that lead to these results. We note again that the Flemish data represents less than a quarter of the CGN dataset. We hypothesize that the high WER

and CER values here are also caused by the limited amount of data that was available for the training process. Notice that removing the files with overlapping timestamps proved to be a better noise handling strategy compared to the removal of face-to-face speech and the spontaneous dialogues (i.e., Non Overlapping versus Small). However, the best results of 38.6% WER and 23.1% CER were obtained when combining the two noise handling strategies.

Table 8 presents the same results on the Dutch data. We note that due to a higher amount of data available for training, the performance of our best model has increased to 34.2% WER and 20.5% CER. Notice also that for this dataset, removing face-to-face speech and spontaneous dialogue proves to be more efficient than removing the files having overlapping transcriptions. After a short analysis, we observed that this is due to a smaller number of overlapping timestamps in the Dutch dataset.

Table 8: Best performing models on the Dutch datasets.

	Nodes	**Learning Rate**	**Dropout**	**WER**	**CER**
Small Non Overlapping	2048	0.00055	0.35	0.342087	0.205417
Non Overlapping	2048	0.00010	0.50	0.466391	0.277180
Small	2048	0.00055	0.20	0.419772	0.273178

Finally, looking at Table 9, we remark that having more data proves really beneficial for our case, obtaining in the end a performance of about 30% WER and 17.2% CER on the combined dataset, when using both noise handling techniques. We also note here that the best performing models for the *non overlapping* and *small* datasets had 1024 hidden nodes per layer. The experiments on these datasets were solely ran on 512 and 1024 hidden nodes per layer, due to only having a marginal increase going from 1024 to 2048 for all other experiments.

Table 9: Best performing models on the combined datasets.

	Nodes	**Learning Rate**	**Dropout**	**WER**	**CER**
Small Non Overlapping	2048	0.0001	0.20	0.299879	0.172096
Non Overlapping	1024	0.0001	0.35	0.408456	0.261991
Small	1024	0.0001	0.20	0.460147	0.326860

5.3 Transfer Learning from English to Dutch

In order to create the best possible models for Dutch, we have attempted to apply transfer learning, by fine-tuning a well performing English model on our

datasets. The model that we have selected was provided in Project DeepSpeech and has a WER of 8.22% on the LibriSpeech clean test corpus. All experiments for transfer learning were conducted using the small non overlapping dataset, which gave the best performing models for all previous experiments.

In these experiments we have trained each neural network in cycles of five epochs, evaluating the resulting network after every cycle. In the following cycle, the network resumed training from the best checkpoint of the previous cycle. This was done for 5, 10, 15, 20, 25 and 30 epochs. We have also tried to push the best performing networks for 40 epochs, but noticed a decrease in performance which can most likely be attributed to overfitting. To obtain a baseline for the performance of the English neural network without any fine-tuning we have evaluated this neural network (combined with the Dutch M-gram language model) on the test sets of Flemish, Dutch and the combined languages. We can clearly see in Table 10 that we obtain a poor initial performance.

Table 10: Performance of the English model (combined with the Dutch n-gram language model) on our datsets before fine-tuning.

	WER	**CER**
Flemish	0.978428	0.611835
Dutch	0.965037	0.631461
Combined	0.972322	0.624038

Table 11 presents the results of our fine-tuning process for Flemish, Dutch and the combined data. We notice a large boost in performance, especially for the Flemish and combined datasets, which reach a WER of 22.9% and 23.6% respectively. This result emphasises the importance of having well-curated and large enough datasets, when training neural STT models. Furthermore, we can conclude that the close relationship between English and Dutch allows us to leverage the features learned while training on the English dataset for our Dutch models.

Table 11: Best performing models with transfer learning.

	Epochs	**Learning Rate**	**Dropout**	**WER**	**CER**
Flemish	20	0.0001	0.15	0.229560	0.133830
Dutch	30	0.0001	0.20	0.255273	0.145501
Combined	30	0.0001	0.20	0.235773	0.134833

6 Conclusion

In this work we have explored how well speech-to-text neural models for Dutch can perform on the Corpus Gesproken Nederlands. To this end we used the architecture proposed by the DeepSpeech project, which we tuned for our purposes. Because the CGN was not directly usable for this task, we provided a full processing pipeline that also includes two noise handling techniques. We evaluated these approaches, by training from scratch models for Dutch, Flemish and a combination of the two. We conclude that, in most cases, removing the files with overlapping transcriptions was more efficient than removing face-to-face speech and spontaneous dialogue, however the best results were obtained when combining both techniques. For models trained from scratch, our best performing models reached a WER of 30.0% and a CER of 17.2%.

A second step in our quest to obtain good quality STT models for Dutch was the attempt to fine-tune a high-performing English model for our dataset. Using this approach we obtained a boost in performance, reaching a WER of 23.0% and CER of 13.4%. This is a strong indication that in the absence of a large volume of high quality data for a specific language, transfer learning could be used to enable successful training on smaller datasets.

For the future, we are interested in further improving and curating Dutch datasets for speech-to-text. This includes developing further techniques for handling noisy data, but also including other smaller datasets such at the one available through the Common Voice project. We are also interested in developing different neural architectures and also using other frameworks such as PyTorch-Kaldi for this task.

References

1. Amodei, D., Ananthanarayanan, S., Anubhai, R., Bai, J., Battenberg, E., Case, C., Casper, J., Catanzaro, B., Cheng, Q., Chen, G., et al.: Deep speech 2: End-to-end speech recognition in english and mandarin. In: International conference on machine learning. pp. 173–182 (2016)
2. Chung, J.S., Nagrani, A., Zisserman, A.: Voxceleb2: Deep speaker recognition. In: INTERSPEECH (2018)
3. Demuynck, K., Puurula, A., Van Compernolle, D., Wambacq, P.: The esat 2008 system for n-best dutch speech recognition benchmark. pp. 339 – 344 (01 2010). https://doi.org/10.1109/ASRU.2009.5373311
4. Demuynck, K., Roelens, J., Compernolle, D.V., Wambacq, P.: Spraak: An open source" speech recognition and automatic annotation kit". In: Ninth Annual Conference of the International Speech Communication Association (2008)
5. Draper, M., Calhoun, G., Ruff, H., Williamson, D., Barry, T.: Manual versus speech input for unmanned aerial vehicle control station operations. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting. vol. 47, pp. 109–113. SAGE Publications Sage CA: Los Angeles, CA (2003)
6. Duchateau, J., Uytsel, D.H.V., Hamme, H.V., Wambacq, P.: Statistical language models for large vocabulary spontaneous speech recognition in dutch. In: Ninth European Conference on Speech Communication and Technology (2005)

7. Durling, S., Lumsden, J.: Speech recognition use in healthcare applications. In: Proceedings of the 6th international conference on advances in mobile computing and multimedia. pp. 473–478. ACM (2008)
8. Evers, J.: Using lstms to improve dutch language models (2017)
9. Fagnant, D.J., Kockelman, K.: Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. Transportation Research Part A: Policy and Practice **77**, 167–181 (2015)
10. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. MIT press (2016)
11. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd international conference on Machine learning. pp. 369–376. ACM (2006)
12. Graves, A., Jaitly, N.: Towards end-to-end speech recognition with recurrent neural networks. In: International Conference on Machine Learning. pp. 1764–1772 (2014)
13. Graves, A., Mohamed, A.r., Hinton, G.: Speech recognition with deep recurrent neural networks. In: Acoustics, speech and signal processing (icassp), 2013 ieee international conference on. pp. 6645–6649. IEEE (2013)
14. Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., et al.: Deep speech: Scaling up end-to-end speech recognition. arXiv preprint arXiv:1412.5567 (2014)
15. Heafield, K.: KenLM: faster and smaller language model queries. In: Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation. pp. 187–197. Edinburgh, Scotland, United Kingdom (July 2011), `https://kheafield.com/papers/avenue/kenlm.pdf`
16. Hernandez, F., Nguyen, V., Ghannay, S., Tomashenko, N., Estève, Y.: TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation. In: International Conference on Speech and Computer. pp. 198–208. Springer (2018)
17. Huijbregts, M., Ordelman, R., de Jong, F.M.: A spoken document retrieval application in the oral history domain. In: Proceedings of 10th international conference Speech and Computer, Patras, Greece (SPECOM 2005). vol. 2, pp. 699–702. University of Patras/WCL Moscow State Linguistics Uni. (2005)
18. Iakushkin, O., Fedoseev, G., Shaleva, A., Degtyarev, A., Sedova, O.: Russian-language speech recognition system based on deepspeech (2018)
19. Jadoul, Y., Thompson, B., de Boer, B.: Introducing Parselmouth: A Python interface to Praat. Journal of Phonetics **71**, 1–15 (2018). https://doi.org/https://doi.org/10.1016/j.wocn.2018.07.001
20. Kessens, J., Van Leeuwen, D.: N-best: The northern- and southern-dutch benchmark evaluation of speech recognition technology. vol. 2, pp. 1354–1357 (01 2007)
21. López, G., Quesada, L., Guerrero, L.A.: Alexa vs. siri vs. cortana vs. google assistant: a comparison of speech-based natural user interfaces. In: International Conference on Applied Human Factors and Ergonomics. pp. 241–250. Springer (2017)
22. Nagrani, A., Chung, J.S., Zisserman, A.: Voxceleb: a large-scale speaker identification dataset. In: INTERSPEECH (2017)
23. Noyes, J., Frankish, C.: Speech recognition technology for individuals with disabilities. Augmentative and Alternative Communication **8**(4), 297–303 (1992)
24. Oostdijk, N.: The Spoken Dutch Corpus. Overview and First Evaluation. In: LREC (2000)
25. Ordelman, R., Ordelman, R.: Dutch Speech Recognition in Multimedia Information Retrieval. Ph.D. thesis, Netherlands (10 2003)

26. Panayotov, V., Chen, G., Povey, D., Khudanpur, S.: Librispeech: an asr corpus based on public domain audio books. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5206–5210. IEEE (2015)
27. Pelemans, J., Demuynck, K., Wambacq, P.: Dutch automatic speech recognition on the web: Towards a general purpose system. In: Thirteenth Annual Conference of the International Speech Communication Association (2012)
28. Pratap, V., Hannun, A., Qiantong Xu, J.C., Kahn, J., Synnaeve, G., Liptchinsky, V., Collobert, R.: wav2letter++: The fastest open-source speech recognition system. arXiv preprint arXiv:1812.07625 (2018)
29. Ravanelli, M., Parcollet, T., Bengio, Y.: The pytorch-kaldi speech recognition toolkit. In: In Proc. of ICASSP (2019)
30. Röpke, W.: Building a Speech-to-Text Engine for Dutch. Bachelor thesis, Vrije Universiteit Brussel (2019), `https://ai.vub.ac.be/files/Ropke_Bachelor_thesis_1819.pdf`
31. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. International journal of computer vision **115**(3), 211–252 (2015)
32. Shaya, S.A., Matheson, N., Singarayar, J.A., Kollias, N., Bloom, J.A.: Intelligent performance-based product recommendation system (2010), uS Patent 7,809,601
33. Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al.: Mastering the game of go with deep neural networks and tree search. nature **529**(7587), 484 (2016)
34. Steeneken, H.: Potentials of speech and language technology systems for military use: An application and technology-oriented survey. Tech. rep., Human Factors Research Institute, TNO Soesterberg (The Nederlands) (1996)
35. Van Eerten, L.: Over het corpus gesproken nederlands. Nederlandse Taalkunde **12**(3), 194–215 (2007)
36. West, J., Bhattacharya, M.: Intelligent financial fraud detection: a comprehensive review. Computers & security **57**, 47–66 (2016)