Building a corpus

For this project, we decided to adapt the methodology described in the Google AI blog, where their team describes the process of developing a Bangla synthetic voice. This seemed ideal, as the target language is an under-resourced language, such as Grunnegs, our target language.

An algorithm such as the one we chose to use, the Tacotron algorithm, is data-driven. This means that a corpus was needed to be assembled. A corpus comprised of both aural and textual data. Following the experience described in the Google AI Blog, we set out to assemble a corpus of two thousand sentences in the target language. This corpus would be then used to train the algorithm, which in turn would produce a model. The resulting model is a function that, given a certain input, in this case, written Grunnegs (e.g. "Moi, ik bin Tammo"), produces the desired output, which in this case speaks Grunnegs.

Designing the corpus

However, to do this we have to take some considerations. The corpus cannot be assembled haphazardly. Some considerations are strictly related to the kind of data that the algorithm accepts. We could call these technical considerations. Because the algorithm is designed to find patterns between the audio files and the text files, which ultimately produces a model that accurately can process text into intelligible speech. Therefore, the first important consideration is to have the aural data and the textual data aligned, which means that there is a correspondence between the text files and the audio files, that is, that what is said in the audio files is exactly what is written in the text files. This is extremely important as we are going to input this data into the algorithm with the objective to produce a model that can accurately represent the regularities that exist between the spoken word and the written word. There are automated procedures to force align data. However, given the fact that the corpus we collected was relatively small, and forced aligned data still has to be checked, we decided to check the correspondence of the data manually.

For the same reason that we need the aural and textual data to correspond to each other, that is, allowing for the algorithm to establish regularities, there cannot be any background noise, which includes also background music. This might seem trivial, especially if seen from the perspective of human beings, which are very adept at differentiating speech from background noise or music. But a computer, a robot or an algorithm are very different.

There are potentially four standard ways to come up with the data (and of course all possible combinations of these four to different degrees). We need text and the corresponding audio. The first option is to write all the sentences and then record them. This might seem a straightforward option but is problematic, as it is uncertain if the text produced is representative of the current language use of the community. A second possibility is to find existing pairs, for example, audiobooks are great. Most under-resourced languages do not have these or are very limited. Sometimes these exist but are not easily available, even when they are in the hands of

public institutions. This is actually the case with our project. We were made aware of the existence of audiobooks recorded in Grunnegs in the ‘90s, but the public institution that holds these recordings has decided to constantly ignore our requests. A third option is to use radio recordings or TV recordings. This is usually referred to as found data. However, transcription can be time-consuming, and there is no guarantee that there are no noises in the background of the audio that can be detrimental to the training process. A fourth option is to, based on texts found (e.g. on the internet), record the audio files. This is, incidentally the path we chose.

Due to the fact that Python can only deal with 16-bit audio files, this is the bitrates needed for the audio files. This means that either the audio files have to be recorded with this bit-rate, or they have to be subsequently transformed from any other bit-rate (e.g. 32-bit) to 16-bit. Additionally, the algorithm only works with tracks recorded mono, so if the audio files were recorded in Stereo, these have to be modified. A way to do this efficiently is to use software such as Adobe Media Encoder to do so, which might take from a couple of minutes to a couple of hours depending on your hardware and size of the corpus.

There are also linguistic considerations to be taken. Since the algorithm is designed to build a model that reflects the regularities between text and audio the corpus has to be as homogeneous as possible. This means that the corpus assembled should be composed of a single speaker recording a single variety. Of course, the speaker can be proficient in multiple varieties, but during the recording of the corpus, it should all be as homogeneous as possible. Or, if it is found data, the audio files selected should all be of one variety. We chose to work with Hogelandsters because we got in touch with a potential speech donor who had previous experience recording audio-books in Grunnegs. We then selected from the corpus we had assembled, texts that reflected the spelling of this variety.

Regardless of whether the corpus was assembled by crawling the internet for texts or whether the texts were written by the team, it is necessary to use a consistent orthography. This is extremely important, not because of some prescriptive obsession, but because the algorithm will be looking for regularities. Therefore, unnecessary variation in aspects of the dataset that could be homogeneous (and the algorithm expects to present a certain uniformity), will result in a low-quality model, which will very likely perform poorly when synthesizing speech. We decided to record texts we retrieved from the website of a literary journal. The use of the literary journal was beneficial, as editors tend to ensure a certain degree of uniformity in spelling, which is crucial in this case, as the algorithm is set to find regularities between the spoken and textual corpora one assembles.

A crucial aspect is to get in contact with potential “speech donors”. To do this it is important to survey the speech community and look for potential institutions that might have members that might be willing to collaborate with the project. We approached several theatre associations which replied positively, although only one had someone available that was both willing to collaborate with our project and was available. In case of having multiple candidates, carrying

out a short interview to ensure that we can choose the person that best fits the desired profile is an advisable step to take.

Recording the data

As mentioned in the previous section, to retrieve the necessary data to train the models needed for speech synthesis there are namely two sets of data that have to be produced: a written corpus and spoken corpus. Ideally, these two parallel corpora are available in the form of audiobooks, accompanied by the digitized texts.

However, under-resourced languages such as is the case of Grunnegs at the moment, do not have the available infrastructure nor the existing resources to develop neither speech technology nor to localize software. The lack of resources means that we had to produce (at least) part of the data ourselves. Following the example of the experiment carried out at Google, and described in the Google AI Blog, recording the audio ourselves on the basis

To produce the required audio files, we recorded at the studios at the Faculty of Arts and at Campus Fryslân. Ideally, it has to be a room without noise, which also means no echo (an-echoic studios).

We used Adobe Audition to edit the audio files (Audacity is also a possibility). To record, we used SpeechRecorder (Clark & Bakos, 2015) developed at the University of Edinburgh. There are other options, such as Praat, which with a plug-in developed by Wilbert Heringa presents similar functionalities to SpeechRecorder. SpeechRecoder allows to efficiently streamline the process of recording, automatically generating unique clips of each recording with a unique name and the corresponding text.

It took about 10 studio hours to record what resulted in 1,5 hours of material. Once the recordings were done, all the data had to be prepared for the next stage. We also recorded the same amount of data in Dutch to use as a control in our experiments.

To record the Grunneger corpus we got in contact with a woman of about 60 years old, who has lived her whole life in the northern area of the province of Groningen where Hogelaandsters is spoken. She identified herself as a speaker of this variety.

The first session of recordings took place at the Faculty of Arts of the University of Groningen on August 6th 2019. The second recording session took place on August 14th 2019. The audios in these sessions were recorded in a proper studio, so there is no echo in the recordings, and the "speech donor" was in a soundproof room, with whom I could only communicate by pressing the talk-back button on the console. In these first sessions, the audios were recorded in batches of fifty sentences, which were presented to the donor one sentence at a time on a PowerPoint presentation, while the audio was recorded using Adobe Audition. This is a licensed program. A suitable option would have been to have used Audacity. This software is available on the internet to be downloaded free of charge.

Additionally, for our experiment we also needed a control condition, so we did a similar procedure to record a Dutch corpus. While the project was advancing, we still researched existing software that might streamline recordings in a better way. Therefore to assemble the Dutch corpus we decided to use SpeechRercorder, software specially designed to assemble corpora that has speech synthesis for its objective. The advantage of this software is that it significantly reduced the amount of time that has to be destined to ready the data for the training stage.

The "speech donor" for the Dutch corpus was a woman of about 70 years old, who was born and raised in Amsterdam. The recordings took place at the Studio in the faculty building of Campus Fryslân, in Leeuwarden. This studio has been intended to film videos, and the acoustics are not ideal for audio recordings, but since the donor lived in Leeuwarden, this studio provided a noiseless environment were to carry out the recording sessions. Because of the availability of the donor the sessions were much shorter than the two full-time sessions carried out to assemble the Grunneger corpus. Recordings took place on February 17th and 19th, and March 2nd, 5th, 9th and 12th 2020.

Once the data had been collected, we had to run a quality check, which basically meant listening again to all audios and make sure that what was being said in the audios was exactly what was written in the text files. Part of this process was being done while the recordings are taking place, as while the donor is reading out loud the sentences that appear on the screen, the person responsible for the recordings should be controlling this correspondence by wearing headphones and simultaneously listen to the input while it is being recorded.