Designing and evaluating methods for the detection of mistakes in Sprekend Nederland data

Author:
Maurits Cornelis Ariëns Kappers
3698890

First supervisor:
Prof. dr. David van Leeuwen

Second supervisor:
Dr. ir. Gerrit Bloothooft

A thesis presented for the degree of Master of Science

Artificial Intelligence

Universiteit Utrecht

Faculty of Science
Utrecht University
The Netherlands

June 1, 2017

Abstract

Due to a constant increase in audio and video information, the demand for methods that automatically align and filter this information keeps growing. Towards that end, this thesis aims to fulfill two objectives. First, to show the accuracy of the Spraaklab aligner and use it to align the Sprekend Nederland corpus, so that it can be used in further research. Second, to design and evaluate a method for automatically detecting user made mistakes in read speech using Spraaklab's alignment process. The Corpus Gesproken Nederlands is used to develop alignment accuracy and mistake detection benchmarks. Spraaklab is shown to be accurate in aligning Sprekend Nederland and the read speech data is aligned. A mistake detection method using word recognition scores is developed, and shown to be effective on the Corpus Gesproken Nederlands. Due to score calibration problems it can not be shown to be effective on Sprekend Nederland, but the results indicate that further research could be able to show it, given more manually verified Sprekend Nederland alignments to establish better thresholds.

Contents

1 Introduction

In an age where digital information is overflowing it becomes increasingly important to have methods that are able to efficiently index and search information, including audio information. In order to achieve this methods have been researched and developed: speech recognition, a method that translates speech in a recording into a textual representation, is useful for transcribing audio; and forced alignment, which aligns a textual representation of the audio in time to the audio, is useful for searching information audio or video segments. The segmentation is a representation of the audio signal in which the content is split according to speech sounds or words. The result of this process allows the content to be searched based on what was said, without needing someone to watch or listen the entire segment to manually map the text to the segment. It is a technique that is already used in searches by video service providers over the world, most notably YouTube, which already uses automatic speech recognition to automatically provide captions for uploaded videos, or aligns the captions for you if the transcripts for said videos are provided [1].

The ability to automatically segment audio based on sounds or words also greatly improves the ease with which languages can be studied and charted, which is the purpose of the Sprekend Nederland project: A large Dutch national research project which intends to create a blueprint of the Netherlands which contains all the different Dutch accents and dialects, and to study the mutual judgments of dialect speakers on their speech. [2].

The research described in this paper has two objectives. To align the currently available data in the Sprekend Nederland (SN) corpus using the Spraaklab aligner, developed by NovoLanguage, and to develop a method by which the aligner can automatically detect mistakes in either the recording or its textual representation.

To provide a basis for the evaluation of the SN alignment a different corpus, the Corpus Gesproken Nederlands (CGN) which already contains manually verified alignments, will first be aligned by Spraaklab. The CGN alignments by Spraaklab will then be used to calculate and optimize the alignment quality of Spraaklab by comparing them to the manually verified alignments in the corpus, and the resulting alignment accuracy will then serve as the gold standard for the expected accuracy on other corpora. Whether this is an accurate assumption will then be verified by aligning SN and calculating the actual performance.

If the accuracy turns out to be satisfactory, in other words comparable to the performance of Spraaklab on the CGN, it is then possible to pursue the second objective of this paper: to investigate whether the process Spraaklab uses to produce alignments can be used to detect mistakes in either the recording or its transcription. By manually introducing mistakes into the CGN alignments it should be possible to develop a heuristic by which recordings with mistakes can be recognized automatically. Finally, whether this heuristic performs well will be tested by measuring its accuracy on SN.

2 Background

This chapter will provide some background information on the methods and data presented in this thesis. Sprekend Nederland, Spraaklab, and the Corpus Gesproken Nederlands will be discussed in more detail: a short history, the purpose, how they are built, and what they contain.

The Spraaklab results differ in structure from the CGN data, so both the CGN and Spraaklab data will need a conversion to a data structure which will allow for the data to be compared. Using the comparison as basis, the Spraaklab parameters will be adjusted for increased accuracy. The SN audio fragments that have transcriptions will be aligned and converted to the data structure designed and used in the first phase. The performance of Spraaklab on the SN data will be calculated and cross referenced with the performance measures resulting from the first phase in order to ascertain whether the performance on SN is consistent with the performance on the CGN.

2.1 Sprekend Nederland

A single country can have many different accents, based on region, city, or even districts within cities. A geographical map of these different accents could be used for linguistic, psychological and sociolinguistic research. In order to map these different accents in the Netherlands the NTR, a public-service broadcaster, in collaboration with the NWO, the Netherlands Organisation for Scientific Research, has launched a large ongoing research project called Sprekend Nederland [2]. As previously stated, the purpose of SN is to make a blueprint of Dutch accents and dialects.

To collect the necessary data for this project an Android App and an iOS App were launched at beginning of december 2015, also called Sprekend Nederland [3]. The app presents the user with a number of categories (e.g., housing, birthdays, and music) within which the user must either answer questions about an audio fragment recorded by someone else, or record an audio fragment him or herself. Most of the categories are unavailabe at the start, but by answering enough questions in one category the user can unlock the next category (the housing category provides access to the birthday category for instance). The questions concern the accent and the users' perception of it. An example of a few questions would be whether the recorded person lives near them, whether they sound sexy, and whether they make a lot of money. The recording sections consist of the user either describing something freely, an image for instance, or recording a sentence or word that is displayed on screen.

For this research, we will only be using the fragments for which the user was asked to record a specific sentence or word. Allowing us to measure the forced alignment accuracy of Spraaklab.

2.2 Spraaklab

Spraaklab is a speech recognizer developed by NovoLanguage, a Dutch company which specializes in teaching languages using technical innovations [4]. Spraaklab is based on the SHoUT toolkit, developed by Marijn Huijbregts for his dissertation as an automatic speech decoding system for unknown audio conditions. It includes: speech/non-speech segmentation, clustering, and automatic speech recognition (ASR). The segmentation separates the fragments that contain speech from those that do not, the clustering groups the speech fragments per speaker (allows for the possibility of unsupervised adaptation), and the ASR subsystem does the actual decoding [5]. Spraaklab uses the decoder provided by the SHoUT tookit.

Spraaklab also uses Kaldi, an open source toolkit for speech recognition research, to process the audio. Development on Kaldi started in the Czech republic, to improve upon a previous project based on HTK and also due to a lack of open source toolkits with a finite-state transducer (FST) based framework. Development was aimed at research in particular and the toolkit was designed in a way that makes extension easier. Kaldi is licensed under Apache v2.0, one of the least restrictive possible licenses. To make use of the functionalities available in the Kaldi toolkit the command line can be used, which means that (shell)scripts also have access to Kaldi's features, and having been written with extensibility in mind, it supports a broad range of approaches on almost every level of the speech recognition process: feature extraction, acoustic modeling, phonetic decision trees, language modeling, and decoders [6][7]. Kaldi was used to develop the acoustic models that are used during this research. The Kaldi functionalities used by Spraaklab are the feature extraction and phonetic desision tree generation. These functionalities, combined with SHoUT's lightweight decoder and an acoustic model, allow Spraaklab to quickly generate forced alignments.

Figure 1 contains an example Spraaklab language model (grammar) of a three word sentence. Between the words of the grammar there is a possibility for the decoder to recognize a silence, if adding the silence will maximize the overall likelihood of the grammar (silences at the start and end of a sentence have a different symbol, but are still optional). This process happens automatically, however, there is a way to influence the probability of silence insertion. The Spraaklab decoder provides the option to set a silence parameter. The silence parameter has a direct effect on the likelihood that a silence is inserted between two words, it affects the probability that a sequence of decoder frames is 'silence'. This parameter does not change the likelihood that a single frame is silence, but rather the likelihood that a section will be recognized as silence. A positive value means an increase in likelihood that silence is inserted, while a negative value means a decrease in likelihood. This also means that, while the silence parameter will have a small effect on whether sounds are recognized as silence, it will mainly increase or decrease the number of silences inserted into the alignment.

Spraaklab can also make use of a lexicon service when aligning a text to its audio counterpart. The lexicon service converts the words of the texts to

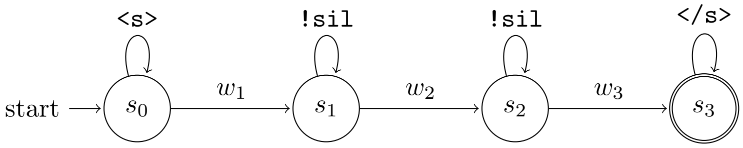


Figure 1: Example Spraaklab grammar

their phone transcriptions which can then be fed to the recognizer. The lexicon service will initially try to look up the word in a dictionary generated from the CELEX corpus, if this does not return a result the lexicon service will generate a phone transcription based on grapheme-to-phone rules learned from the CELEX corpus.

2.3 Corpus Gesproken Nederlands

The Corpus Gesproken Nederlands is a database for contemporary Dutch as spoken by adults in the Netherlands and Belgium [8]. The aim of the CGN project, which ran from 1998 to 2004, was to construct a database for the Dutch spoken language. The corpus itself consists of approximately 10 million words, two-third of which are spoken by readers from the Netherlands, the other third is Flemish, as spoken in Belgium. It is a high-quality corpus and has featured in many scientific papers and journals [9]. All of the recordings in the CGN have alignments provided, and of these forced alignments 10% have been manually verified and corrected. This manually verified section of the CGN is known as the core corpus.

The core corpus will be used as the gold standard to compare with when measuring and optimising the Spraaklab performance. Because the CGN core corpus has been manually verified and corrected we can be sure, barring human error, that the core corpus consists of accurate and correct segmentations and alignments. It is important to note that the manual verications only go down to word level, so this research will also measure accuracy down to word level, since there is no gold standard for phones that can be compared with.

The CGN as a whole consists of fourteen different sections, containing dialogues as well as as monologues. Each of these sections encompasses a different type of speech (e.g., phone conversations, or recorded lectures), but the section that will be used for this research is section o: "read aloud text". It contains read speech, making it comparable to the way in which SN requires its users to record speech. The sections contain six types of annotations to account for different types of research: orthographic transcriptions, part-of-speech tagging, lemmatisations, lexicon link-ups, broad phonetic transcriptions, and word segmentations (also containing the alignments). The CGN also has a lexicon which provides lexical information for the words present in the corpus.

Of these types the most important for this thesis are the word segmentations, which contain a verbatim transcription of the recording.

3 Measuring Spraaklab performance on the CGN

In 2004, Chen et al. evaluated factors impacting the accuracy of forced alignments. In contrast to many previous studies which focused on word error rate and phonetic alignment accuracy they focused on word alignment accuracy. The goal of their research was to understand the factors contributing to accurate forced alignments, in order to minimize the time needed to manually correct the alignments. In order to measure the accuracy of the alignments produced they devised an evaluation method which measured the shift in word boundaries between the obtained results and their gold standard. One of the findings they made was that overall aligment accuracy rose when they initially segmented the speech files based on silences of half a second or more [10].

In 2014 Strunk et al. used a very similar technique to evaluate untrained forced alignments of endangered languages. Referring to segmented data as 'constrained', providing the recognizer with time constraints within to search for specific words, and referring to data that was not initially segmented as 'unconstrained, simply providing the audio data and its transcription. The constrained time markers are based on coarse segmentations done during the transcription process in many corpora, usually aligning a larger annotation unit, such as sentences or paragraphs, to the audio. Since they had no access to an independently created gold standard they evaluated based on a more practical approach: the number of clearly misaligned words that had to be manually corrected. They also found that constrained alignment contributed to a large increase in forced alignment accuracy [11].

Silence also has a large impact on alignment accuracy, unfortunately silence is not as straightforward a problem to solve as one would think. In speech the accepted convention is to use a three-state representation for events: silence, unvoiced speech, and voiced speech. During silence there is no speech, and the audio usually consists of background noise. Voiced and unvoiced speech differ in the representation of their speech signal. Voiced speech has a periodic waveform, thus making it easy to mark as speech (vowels are typical voiced speech). Unvoiced speech has a more random-like nature though, making it harder to classify as speech (many fricatives, and most plosives are unvoiced). Because of the random nature of unvoiced speech it can confuse the decoder and be classified as silence, making it one of the more general problems of silence in speech recognition [12]. A speech activity detector (SAD) can be used to classify speech/non-speech sections (Kaldi does not have a built-in SAD but does allow for integration of one), but SAD performance goes down significantly as background noise increases [13]. Because silence has so much impact on alignment accuracy it is also an important factor in this research, and will be handled accordingly as evidenced by the analysis methods that focus specifically on the effect of silence on the alignment accuracy when compared to the general alignment accuracy.

3.1 Methodology

The methods used to evaluate the accuracy of Spraaklab will be largely based on the papers mentioned in the previous section, and will be expanded with a larger attention to silences. Previous research found that initial segmentation of the audio based on provided annotation units in the corpus increases the alignment accuracy drastically, therefore the data in this research is initially segmented based on the sentence-like structures provided in the CGN. This works in favor for the rest of the research as well, since the SN data only consists of sentences and words.

The specific data we looked for while evaluating were the following:

- **Word Begin Boundary Shift (WBBS)** Will be calculated by taking the begin time of the word in the Spraaklab results and then subtracting the begin time of the same word in the CGN alignment.
- **Word End Boundary Shift (WEBS)** Will be calculated in the same way as the WBBS, this time taking the end time of the word.
- **Shift as a Function of Word Length** Will provide a more detailed analysis of the boundary shift, specifically as a function of word length. The boundary shift should definitely not be larger than the word it belongs to.
- **Word Length Comparison** Simple comparison of the word lengths of both sources, will tell us whether Spraaklab shortens or lengthens words. Ideally the lengths of corresponding words are the same.
- **WBBS and WEBS and Silences** We will separately analyze boundary shifts adjacent to silences, to see whether these have a large impact on the shift. Reason for this is that many recognizers handle silences differently.
- **Silences per Sentence** Spraaklab allows you to set a silence parameter, impacting the probability that a silence is inserted before, after, or in between words. This setting will almost certainly have an impact on the accuracy of the alignment and therefore will be used when processing the audio to determine how much of impact this has on the alignment accuracy.

Together, these analyses will provide a measure of the accuracy of Spraaklab when aligning a corpus such as the CGN. This accuracy also provides us with a benchmark when evaluating the SN alignments generated by Spraaklab.

3.2 Data

The CGN and the results of the Spraaklab recognizer cannot be compared as is since they are structured differently. The CGN corpus data comes in the form of XML files, and the results of Spraaklab are constructed of JSON objects. In order to process the data from both sources the choice was made to create a data structure in Python. Python was then used to extract the relevant information from the CGN, and these data extractions were then partially used to generate the alignments with Spraaklab. To analyse the converted data it will be exported to CSV files which are then processed with R, a statistical computing language.

The CGN file structures happens to be structured in a way that focuses on sentences and the words contained within. This is consistent with the way that the SN data is provided: fragments based on sentences, or isolated words. More specifically, the CGN is split into fragments which cover a 'speech-section', for instance a single conversation or a book chapter read out loud. The fragments are split into sentence-like structures, the annotation unit used for presegmentation, and further split into words.

The data structure used in this research is based on the CGN file structure, making it easier to extract the necessary information from the CGN, to process the sentences with Spraaklab, and to evaluate the way in which Spraaklab will process the SN data.

3.2.1 The unifying data structure

In designing the data structure used for comparing the data from the CGN and Spraaklab it was necessary to take the structure of the CGN files into account without leaving out the essential data needed for the Spraaklab evaluation. Looking at the methodology that we will be using when evaluating it is obvious that the data structure needs constructions for fragments, sentences and words, including begin and end times, and references to their respective counterparts from either the CGN or the Spraaklab results. We also need a way to save silences and their respective places within the sentences.

The next section will cover the structure of the CGN annotations in more detail. To make it easier to process on a per fragment basis the data structure designed for this research allows for fragments to be entirely encompassed, retaining chronological order of the sentences (and words). The research data structure is described in table 1.

It should be noted that silences do not have a separate Python class, for the simple reason that they can be encompassed very intuitively within a word object. Silence is the absence of words, so we chose to represent silence by storing the symbol `<silence>` in the word reference, and leaving the word transcription empty. This allows us to easily recognize the silences present in order to process them, and still contain them within the same data structure.

The structure itself does not contain methods for analyzing the data, it is purely meant as a way of converting the data from both sources to a format which provides the user with an easy way to traverse the information in question. Words and sentences can be retrieved based on reference, and although silences cannot be retrieved on reference, all the data is chronologically ordered so we still have all the information needed: location, duration, and number of silences per sentence.

When it becomes necessary to analyze and compare data, the data structure

Object	Element	
Fragment	Reference	Unique fragment identifier
	Sentences	A doubly linked list of processed sentences, can be used to store CGN sentences, Spraaklab sentences, or SN sentences.
Sentence	Reference	Unique sentence identifier
	Words	Doubly linked list of word objects in this sentence
	Start	Begin time of sentence (in ms)
	End	End time of sentence (in ms)
Word	Reference	Unique word identifier
	Transcription	Word transcription
	Start	Begin time of word (in ms)
	End	End time of word (in ms)

Table 1: Research data structure design

allows for easy exportation of the information to a format which can be analyzed with R.

3.2.2 Processing the CGN

The data provided by the CGN consists of the audio fragments and their respective annotation files. The annotation files that contain the information that we need are of the .skp type, containing a "chronological representation of the orthographic transcription in an XML text format" [14]. They are hierarchically structured, starting with the fragment element, followed by sentence elements, which contain word elements. A generalized CGN file design can found in table 2.

```
<ttext ref="X">
  <tau ref="X.1" tb="" te="">
    <tw ref="X.1.1" tb="" te="" w=""/>
    <tw ref="X.1.2" tb="" te="" w=""/>
  </tau>
  <tau ref="X.2" tb="" te="">
      etc...
  </tau>
</ttext>
```

(a) Generalized .skp file content

`<ttext>`	A fragment element
`<tau>`	A sentence element
`<tw>`	A word element
`ref`	Unique element identifier (extended for sentence and word elements so parent elements are traceable)
`tb`	Begin time of the element
`te`	End time of the element
`w`	The orthographic transcription of the word

(b) Element and tag descriptions

Table 2: CGN .skp file design

The elements all contain tags which provide more information about their respective elements, although they actually contain three more tags than described, which we will *not* be using: *s* (speaker), *tt* (timespan), and *tq* (quality of the timespan). The values for these elements will never change for the fragments that we are processing: every fragment has a single speaker, the timespan always coincides with begin and end times of the element, and the timespan quality is always checked manually. In other words, it is not necessary to pay attention to these elements and their values. The format also allows for comment and background information tags, but these are not necessary for processing the data needed for this research and will be skipped during extraction. Silences are not explicitly stated in the CGN files but can be extracted by comparing begin and end times of adjacent elements, if the end time of an element and the begin time of the following element are not equal then the section between these two times is classified as silence [14].

The CGN files are processed using python and each fragment is processed separately. The sentence elements in the files are then traversed chronologically, and the words within them as well. In case of the sentence elements, the word elements for that sentence are saved in a contained array. In case of the word elements, the word transcriptions are saved.

At the end of the process the result is a fragment object containing the CGN fragment extraction, including the necessary information as outlined in the previous section. This fragment object can then be used to generate the Spraaklab alignments, and to retrieve information on specific sentences or words, or to iterate over them in an efficient way.

3.2.3 Generating alignments with Spraaklab

To generate alignments with Spraaklab we need a recognizer object, a transcription to align, and the audio data on which to align the transcription. The transcriptions are provided by the sentence elements in the CGN data we extracted in the previous section, the audio data is provided by the CGN, and the recognizer is an instance of a Pyhon Spraaklab object using a Dutch acoustic model trained on the CGN.

The Spraaklab alignments are generated while iterating over the sentence elements present in the CGN data we extracted. By looping through the data we gain access to the sentence reference, the begin and end times of the sentences, the word elements in the sentence element (and by extension the transcription), and the location of the audio file.

The audio files that come with the CGN consist of fragments which are not split into separate sentences. We will be doing a constrained analysis however, processing each fragment sentence by sentence, necessitating the ability to either split up the audio data, or only processing a part of the data. Spraaklab luckily already possesses functionality to this end so this process was easily automated. The extracted sentence audio coincides exactly with the begin time and end time of that sentence provided by the CGN. These sentences were aligned to start at the beginning of the first word, and end at the ending of the last word, silences being represented as unclassified time (as stated earlier). This means that in the CGN there are no silences at the beginning and the ending of the sentences, meaning that there is a possibility that word boundary shifts of words at the beginning and ending of sentences are skewed towards the sentence, since the boundary generated by Spraaklab cannot go past the start or end of the audio data.

In order to prevent inaccuracies occurring because of these boundaries we will analyze audio sentence data as determined by the boundaries provided by the CGN, as well as audio sentence data of which the begin and end boundaries have been extended, by adding the silent audio already present in the data at the beginning and ending of the sentence element, while also adding the corresponding `<silence>` objects. The extension on both sides has a maximum of 100 ms, or half the length of the silence preceding or following the sentence, preventing sentence overlap.

A generalized form of the JSON results generated by Spraaklab can be found in table 3, it should be noted again that while Spraaklab *does* produce phone-level segmentations, the CGN *does not* have manually checked phone-level segmentations, therefore the phone-level information produced by Spraaklab will not be used.

```
[
  {
    "begin": <word begin time>,
    "end": <word end time>,
    "phones": [
      {
        "begin": <phone begin time>,
        "score": <phone score>,
        "end": <phone end time>,
        "label": <phone label>
      },
      {<phone element>}
    ],
    "label": <word transcription (label)>,
    "score": <word score>,
  },
  {<word element>}
]
```

Table 3: Generalized Spraaklab JSON results

One Spraaklab JSON result encompasses one sentence from the CGN. The information that we need to extract to our own data structure are contained by the *begin*, *end*, and *label* tags. The rest of the information needed, for instance *reference*, can be inferred when cross referencing the Spraaklab results to the CGN data already present in our data structure. When extracting the Spraaklab results to our data, the following elements are needed: begin and end times of words and sentences, the words need to be contained within their sentence objects, the word transcriptions need to be contained in their word objects, silences need to be saved in a usable way. And very importantly, the references to the sentences and the words absolutely need to be correct.

The begin and end times of the elements in the Spraaklab results are represented by the number of frames, small 10ms segments over which features are computed, that the element encompasses. In order to compare this to the CGN data we need to convert them to milliseconds, however, because of the frame size Spraaklab will never be more accurate than 10 ms (it only has information over the entire frame, not the individual milliseconds). Word elements in the results contain their transcription in the *label* tag, so while processing the word boundaries the word transcription can also immediately be extracted. Silences are a bit trickier, Spraaklab has three different symbols that describe silence: `<s>`, a silence which begins the sentence; `!sil`, an inter word silence; and `</s>`, a silence which ends the sentence. The symbols only describe a different position within the sentence, not different types of silences, and their presence is not a requirement (Spraaklab does not require `<s>` and `</s>` to respectively begin or end the sentence). By filtering for these three symbols we can simply add a `<silence>` object with the corresponding begin and end times to our sentence object. This way it can be compared to the CGN silence object, while also conserving chronological order.

In order to insert the correct references we need to know the corresponding sentences and words in the CGN data. The sentence reference is already available since the Spraaklab results are generated on a sentence by sentence basis, while traversing through the CGN sentences. We simply need to copy the CGN sentence reference to the Spraaklab sentence reference. The word references cannot be copied in the same way because the addition of silences means that the sequence of the word objects for the CGN sentence and the Spraaklab does not have to be identical. To remedy this problem we iterate over the sentence while cross referencing the word transcriptions until the correct word object is found. Then the reference from the CGN word is copied to the Spraaklab word.

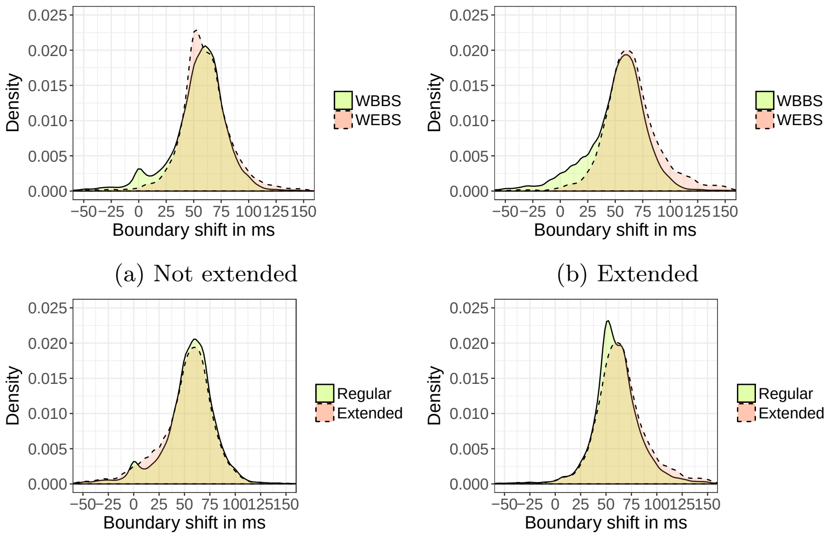
Once all the Spraaklab sentence alignments have been extracted to our data structure we have all the data we need to evaluate the accuracy of Spraaklab based on the evaluation constraints we set at the beginning of this chapter.

3.3 Performance evaluation

The data exported from the processed fragment objects (each type of evaluation had its own exported data to prevent confusion) and was consequently analysed using R and standard analysis methods. Ten fragments were analyzed, containing 436 sentences, and a total of 5162 words.

3.3.1 Looking at the general data

We will start out with the general data, figure 2 contains the *word boundary shift* density plots, whereas table 4 contains a summary of the compared boundary shifts. All shifts are calculated by subtracting the CGN word boundary from the Spraaklab word boundary. There is a clear tendency to the right, meaning that Spraaklab routinely recognizes words "too late". Spraaklab systematically marks the start and end of a word approximately 50/60 ms later than the annotation in the CGN . Spraaklab also makes the sentences longer: when calculating the mean difference between the last boundary of a Spraaklab alignment and the last boundary of its CGN counterpart the difference is 50.17 (an example can be found in table 8 at the end of this section). This should not be possible since the partial audio ends at exactly the end of the CGN annotation, but might also be due to the systematic shift, which has approximately the same duration.



(a) Not extended

(b) Extended

(c) Begin boundaries

(d) End boundaries

Figure 2: Density plots for the general boundary shifts (top plots contain begin and end boundaries, for either extended or regular data, bottom plots contain extended and regular shifts, for either begin or end boundaries)

Boundary	**Min**	**Mean (SD)**	**Max**
Begin	−686	50.18 (43.34)	324
Begin (Extended)	−586	47.22 (44.73)	324
End	−299	61.87 (33.12)	616
End (Extended)	−319	65.92 (34.30)	616

Table 4: Boundary shift summary in ms (calculated by subtracting the CGN word boundary from the Spraaklab word boundary)

Extending the beginning and ending of the sentences with silence improves the shift almost always by less than 10 ms, and since Spraaklab is not accurate below 10 ms because of the frame size used by the decoder this does not improve the overall accuracy. Interesting is how the begin boundary shifts become smaller but the end boundary shifts become larger, meaning that Spraaklab thinks words are longer in the extended test cases (it should be noted that this is a side effect of extending the sentence with silence, and not of Spraaklab assigning more time to a sentence than the CGN.

Even though the boundary shift tendency is not optimal, it is by no means disastrous. More important is that the shifts do not exceed the length of the word associated with them. Should the shift be larger than the word length than it could mean that Spraaklab recognized the end of a word before it even started, or after it ended. Figure 3 shows the boundary shift of the begin and end boundaries plotted as a function of Spraaklab word length. The lines in figure 3 indicate the shift compared to the word length where they are the same duration. Begin boundaries should not be to the left of the dotted line, and end boundaries should not be to the left of the dashed line. As can be seen this is case for most boundaries (not all of them though).

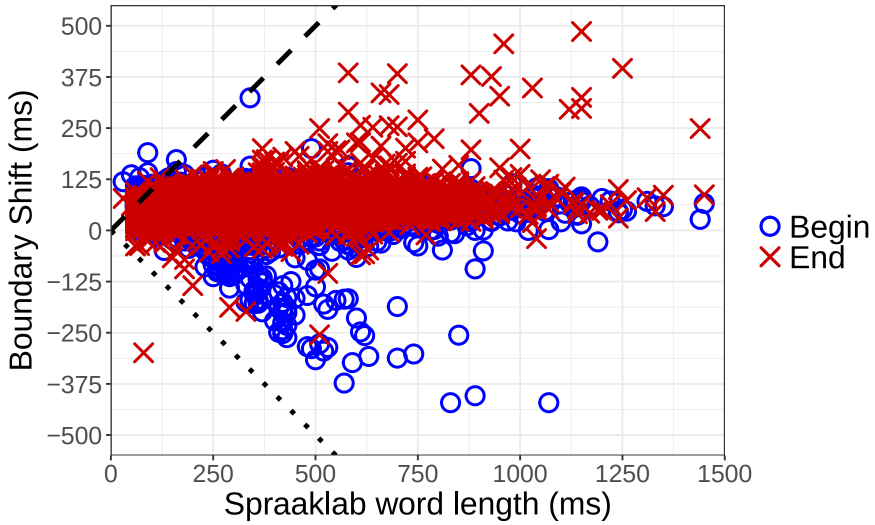
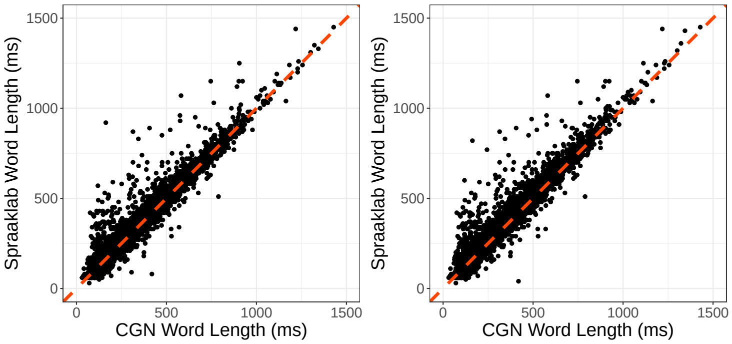


Figure 3: Boundary shift (of unextended data) as a function of Spraaklab word length

Figure 4 shows that the lengths of the words recognized are in most cases approximately the same. Spraaklab has a small tendency to lengthen the words when compared to CGN. The mean word length of the extended data is 337, slightly larger than the mean word length of the unextended data which is 330, which was expected.



(a) Not extended

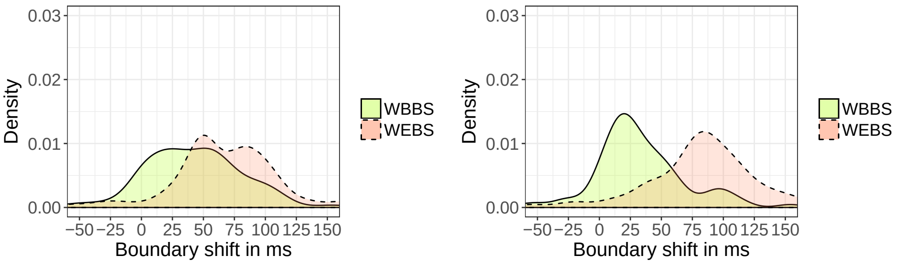
(b) Extended

Figure 4: Word length comparison between CGN and Spraaklab

3.3.2 Looking at the silence specific data

Since silences are interpreted differently by many recognizers, we decided to devote a more detailed analysis to it, in order to see whether silences have a significant impact on the Spraaklab recognizer. Figure 5 is a density plot of the boundaries, regular and extended, filtered on the condition that the begin, or end, boundary is shared with a Spraaklab silence object. Whereas figure 6 is a density plot of the boundaries that are *not* shared with silence objects. Table 5 contains the summary for the data presented in figure 5, and table 6 contains the summary for the data presented in figure 6.

Looking at the distribution it is very obvious that the distributions for the boundary shifts adjacent to silent sections look different from the boundary shifts containing all the boundaries in figure 2, or the boundary shifts not adjacent to silence in figure 6. The begin boundaries move closer to their CGN annotated counterparts, and the end boundaries move further away. This movement is more pronounced in the extended data, probably because the first and last words of the sentences can move past the data audio limit from the unextended data. It is important to note that the distributions in figure 5 consist of very small datasets (171 for the regular data, 172 for the extended data), so a larger set is needed for more accurate conclusions, although the data does seem to provide a trend. On the other hand, the distributions concerning the data of word boundaries *not* adjacent to silence look very much like the distribution in figure 2, this is not surprising, considering that the datasets are much larger (3407 words for the unextended data, 3059 for the extended data), and make up the majority of the data presented in the distributions containing all the boundaries.



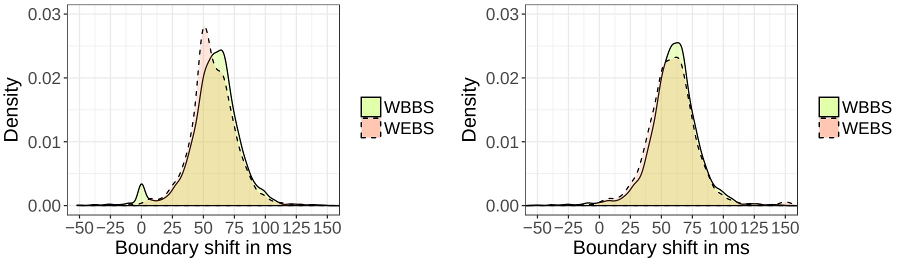
(a) Not extended

(b) Extended

Figure 5: Density plots for boundary shifts shared with silence (including silences shorter than 50 ms)

Boundary	**Min**	**Mean (SD)**	**Max**
Begin	−256	40.49 (53.93)	324
Begin (Extended)	−256	31.08 (53.01)	324
End	−299	66.83 (69.32)	376
End (Extended)	−319	80.28 (69.75)	346

Table 5: Boundary shift summary (boundary shared with silence)



(a) Not extended

(b) Extended

Figure 6: Density plots for boundary shifts not shared with silence

Boundary	**Min**	**Mean (SD)**	**Max**
Begin	−52	58.55 (21.02)	173
Begin (Extended)	−100	59.99 (19.83)	173
End	−52	57.36 (18.91)	200
End (Extended)	−52	58.59 (20.77)	200

Table 6: Boundary shift summary (boundary not shared with silence)

Concerning the number of silences recognized in sentences, there is a clear difference between the CGN and Spraaklab, which can easily be seen in figure 7, containing a plot of the number of silences in a CGN sentence against the number of silences in the corresponding Spraaklab sentence.

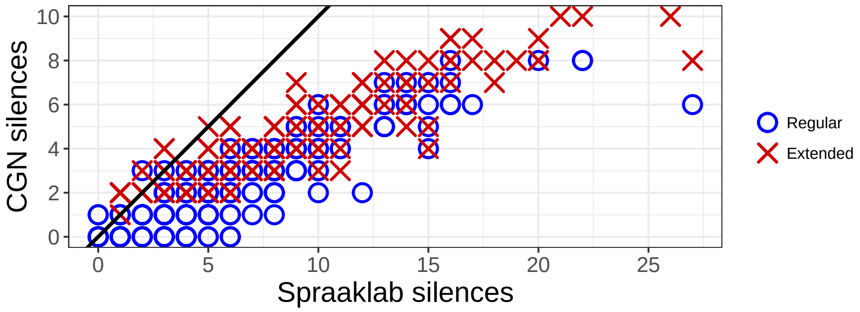


Figure 7: Silences in CGN and Spraaklab sentences plotted against each other

Spraaklab is twice as inclined to recognize a silence when compared to the CGN. To be fair, there are several reasons as to why this number is so high. The CGN protocol, concerning manual alignment correction, tells the correctors that silences shorter than 50 ms have to be removed. However, silences shorter than 50 ms are still present in the Spraaklab data. To be able to present a fairer comparison between the two datasets the Spraaklab data was also post processed to concatenate adjacent silences, and to remove silences shorter than 50 ms. The begin and end boundaries adjacent to removed silences were placed in the center of the silence. Figure 8 contains a comparison between a CGN annotated sentence, and the same sentence processed by Spraaklab, and post processed to concatenate silences and remove silences shorter than 50 ms.

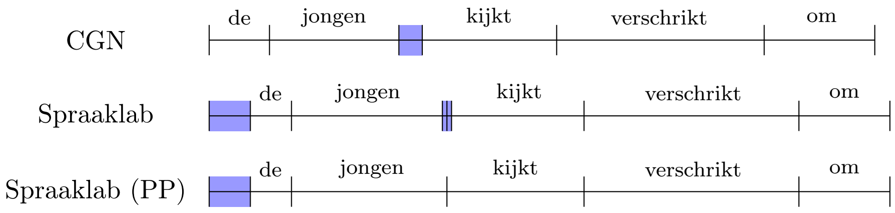
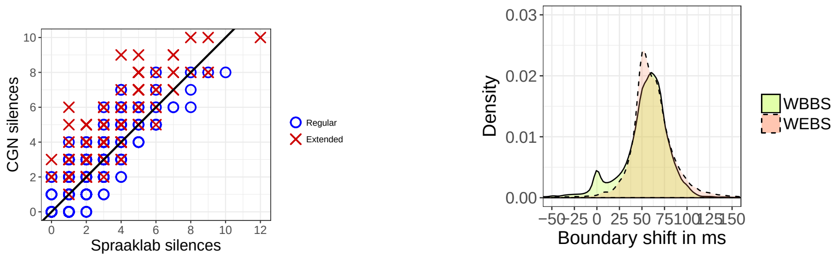


Figure 8: Example of silences in the CGN compared to (post processed) Spraaklab, marked areas are `<silence>`. The reference to the sentence in question is *fn001001.5*

To see whether post processing would have an impact on the accuracy of the alignment, the post processed data was analyzed again. As can be seen in figure 9a the silences per sentence see a major improvement over the regular data, seeing a close to 1-on-1 silence trend, although Spraaklab still inserts more silences. Post processing the data does not, however, have a large impact on the boundary shift distribution (as can be seen in figure 9b), so the larger number of silences in the Spraaklab results is not the reason for the systematic 50/60 ms shift.



(a) Silence count comparison per sentence

(b) Boundary shift distribution (for regular data)

Figure 9: Post processed reference plots

Silence Prior	Boundary	Min	Mean (SD)	Max
1	Begin	−686	50.3 (43.24)	324
	End	−299	61.68 (33.14)	324
−1	Begin	−666	50.1 (43.37)	324
	End	−299	61.9 (32.87)	616
10	Begin	−686	50.91 (43.18)	324
	End	−299	59.88 (34.34)	616
−10	Begin	−686	49.36 (43.65)	324
	End	−299	62.28 (32.84)	616
100	Begin	−686	58.8 (44.78)	324
	End	−299	50.82 (35.60)	616
−100	Begin	−686	45.18 (44.90)	324
	End	−299	62.51 (32.54)	616
250	Begin	N/A	N/A	N/A
	End	N/A	N/A	N/A
−250	Begin	−686	44.53 (46.78)	200
	End	−299	63.31 (34.65)	616

Table 7: Boundary shift summaries by silence parameter value

Finally, table 7 contains the summaries of the boundary shift when processing the data with different silence parameter values. The values in the first columns represent the silence parameter. They were chosen in a growing pattern to research the effect on the alignment accuracy since the posterior for recognizing an audio section as a word does not have a standard value. As can be seen in the table the effect was minimal. The most probable reason for these results is the that the silence parameter does not influence the silence posteriors for individual frames, but rather gives a single bonus for starting a silence section. If the score for a section of two words becomes higher when inserting a silent section in between (even when the scores of the words go down), a silence is inserted. However, once it has been inserted there is no higher score to gain. The silence parameter therefore really only has an effect on the number of silences in the fragment, and not on the length of the silences. In any case, the silence parameter has too little of an effect to truly impact the accuracy of the decoder.

3.3.3 Systematic delay

Since the delay seems to persists no matter the variables and parameters set, the choice was made to take a more in-depth look at some sentences during the entire process of aligning. By looking at the ingoing data and consequently the outgoing results it should be possible to either prove that the delay of 50 ms is systematic, or at least provide a reasonably assumption whether it is or it is not.

Finding out proved simpler than originally estimated. Spraaklab, for reasons unknown, adds 50 ms to the length of the data being processed. An example of this can be found in table 8 (notice the difference between the Spraaklab alignment end time and the other end times). This is both problematic and a relief. It is problematic because the audio data that Spraaklab works with are cut from the fragment, and have a specific duration. Before the first frame, or after the last frame, there is nothing to work with, and yet Spraaklab assumes that the audio data goes on longer than the last frame. Since no exceptions are raised when aligning the sentence it is likely that the 50 ms are added after the actual aligning has taken place, which explains the fact that results are still good. It is also a relief, since this explains the consistent delay present in all alignments. When taking the delay into account the alignment accuracy performs well, 90% of the boundaries placed by Spraaklab differ 50 ms or less when compared to the gold standard boundaries, and in case of the post-processed data, 90% differs 60 ms or less.

It also begs the question if Spraaklab inserts the 50 ms delay at the start of the sentence in the form of a silence, since this seems to be the case in figure 8. When going through the CGN analysis data however, this turns out not to be the case. Of the 436 sentences present in the data processed, only 356 start with a silence after being aligned by Spraaklab, if the 50 ms delay truly was shifted to the front this would have been equal to the total number of sentences. Also, when inspecting the initial silences, the minimum duration is 20 ms, too short to account for the delay. So, although Spraaklab tends to put a silence at the start of an aligned sentence, it is not compensating for the delay at the end.

Source	**Start time (ms)**	**End time (ms)**
Audio data	0	2420
CGN Files	0	2420
Spraaklab results	0	2470

Table 8: Duration data for sentence `fn001001.2`

3.3.4 Conclusion

On average, Spraaklab is 50 milliseconds late when recognizing begin boundaries, and 60 milliseconds late when recognizing end boundaries. The reason for this, as shown in the previous section, is a systematic delay by Spraaklab of 50 ms. The fact that end boundaries have an additional 60 ms delay is probably due to harder to detect consonants. Unfortunately there is no way to locate the source of the delay since we have no access to the source code used by Spraaklab.

Keeping the delay in mind reveals that the Spraaklab aligner is very accurate, with 90% of boundaries differing 50 ms or less from their gold standard counterparts. The comparisons between the word durations of CGN and Spraaklab also confirm this conclusion. Spraaklab does tend to insert a lot of silences in sections where one silence would have sufficed, but, when post-processing the data the amount of silences between CGN and Spraaklab have a close to 1-on-1 ratio. Combined with the fact that silences can simply be filtered during evaluation or future research, the tendency to insert silences does not pose a big problem.

Regarding the silence parameter, and extension of the audio data (using available fragment audio data), there are no significant improvements. From this point on the default settings will be used when processing and aligning data.

Taking all of this into account it can be concluded that Spraaklab is accurate enough to justify aligning SN data, the accuracy can be pinpointed close enough to be able to indicate whether Spraaklab's accuracy on SN is comparable to Spraaklab's accuracy on the CGN.

4 Aligning Sprekend Nederland

Although aligning the SN data is comparable to the method used on the CGN, verifying whether the alignment accuracy is comparable to the alignment accuracy on the CGN will require a different approach then previously taken. The SN corpus does not contain annotated data, which means that we have no gold standard to work with, meaning that an automated verification process is out of the question.

Strunk et al. faced the same problem when evaluating their untrained alignment of endangered languages. Some of the corpora they used had no annotations at all, and some had annotations and segmentations at the sentence level to work with, much like the sentence annotations in the CGN. To calculate the accuracy of the aligner, without a gold standard, they checked the results of the automatic alignment manually while listening to the recordings, correcting clearly misaligned boundaries. Since the corpora were small and they had multiple researchers they were able to process all of the recordings. This method, though not as precise as comparing to a gold standard, is meant to be a practical evaluation to provide information about the percentage of words corrected and the average boundary shift necessary [11].

This method of evaluation is a good fit for the research presented here because it is rather simple. The author has little to no experience in manual word or phone alignment and is the only person performing manual evaluations. The definition of a 'misaligned word' will be constrained to words that are misaligned more than 50 ms in either boundary, this will decrease the number of mistakenly shifted boundaries due to the author not being to pinpoint where one phone ends, and another phone starts, to the millisecond. It should be noted that the sample set being evaluated is set to 100 recordings due the limitations on experience and manpower.

4.1 Methodology

The main goal for this part of the research is to align the SN data provided, and to verify to a certain extent that the alignment is approximately as accurate as the Spraaklab alignment of the CGN. The indicated accuracy should be comparable to the CGN accuracy, when excluding incorrect recordings.

Aligning and verifying the SN data broadly consists of the following steps:

- **Aligning the SN data** The necessary audio and texts are already present. After converting the texts to a format usable by Spraaklab, an alignment can be calculated. No significant changes occurred to the CGN alignments when using values other than the default (0) for the silence parameter, or when the audio was extended to have silences at the start and the end of the sentence, therefore the default python Spraaklab object will be used.

- **Verifying the accuracy of the SN alignment** Two separate methods will be used to evaluate the accuracy of the the SN alignments.

- **Manual correction of SN sample** A sample of 100 sentences was randomly selected from the SN data. The boundaries will be manually evaluated in Praat, and corrected if the author perceives the actual boundary location to be more than 50 ms away. Analysis will be performed on the number of boundary changes, the mean size of the change, and incorrectly recorded sentences.
- **Length distribution of frequent words** Frequent words will be selected based on their phonethic length (Spraaklab does generate phones, but they are simply not used to asses alignment accuracy due to the lack of a gold standard for phones in the CGN), and these will be used to generate duration distributions. Outliers will be checked to see whether the reason for the difference in duration is due to the recording or due to Spraaklab.

4.2 Data

SN has been broadly described at the beginning of the thesis, but it is important to have a more in-depth knowledge of the data present in it. The current corpus available to us contains 217,316 recordings. These recordings can be broadly divided into three sets of recordings: Recordings of a displayed sentence, recordings of a paced set of 5 words (displayed in random order), and spontaneous speech recordings where the user was asked to describe either the word on the screen, or the image displayed. Only the recordings of the displayed sentences will be used here, because they provide the textual representation of what is being said. The corpus has a file containing data on all the recordings, including the transcription and the type of the recording, which allows for easy selection of the sentence recordings. Table 9 contains a general description of the available fields in the recordings file.

Element	**Description**
`rid`	The recording identifier, unique for every recording
`pid`	The participant identifier, all recordings by the same user share this identifier
`file`	Location of the audio data, relative to the audio folder
`text`	Describes the recording, in case of the sentences a transcription, in case of the paced word set the words (but not necessarily in the correct order), and in case of a descriptive task a textual description of what the player is asked to describe
`tgid`	Recording type identifier, this can be used to select only recordings of displayed sentences

Table 9: Available fields per recording in *recordings.csv*

After selection the number of recordings to work with is 112,054. Important to note is that not all the recordings have been correctly recorded. SN is not a carefully crafted corpus like the CGN, but a national project designed to have as many people contribute as possible. Although many recordings are correct, there are also recordings where a user incorrectly reads the sentence, recordings where a user stops recording too soon, or recordings where nothing is being said at all. These incorrect recordings also have to be accounted for when evaluating the accuracy of the alignments.

In order to align the recordings, the same datastructure will be used as was described earlier in table 1, since the datastructure was designed in a way to be able to contain alignments in general. Instead of fragment references to sentences the `rid` will be used, and the word reference is generated in the same the CGN generated the word reference, by appending the word position within the sentence to the sentence reference, starting at 1.

4.3 Processing and aligning SN

Processing and aligning the data from SN is done almost identically to the process used with the CGN, in some ways even simpler. The data used for alignment: the sentence and the audio data (location), can all be found in the recordings file provided with SN. The textual sentence representation needs to be stripped of punctuation in order to look up the phonetic spelling in the dictionary service, and the correct relative folder has to be found. Since there is no gold standard that can be compared with there is also no need to line up the alignments to different data. And since all of the bugs in the processing scripts were already fixed for the CGN data there was not a lot of scripting involved.

Aligning the SN data was much the same as well, with one rather large difference. The SN data to be aligned is much larger than the CGN data used. In order to prevent needless amounts of memory to be used we altered the scripts so that recordings, sentences in other words, can now be saved and loaded individually based on `rid`. No sentence information like begin or end times had to be extracted for alignment since every recording already encompasses a single sentence, and references are available in the recordings file as well. The same acoustic model was used as before.

The alignment process mainly consists of making sure that the correct recordings are aligned, based on the `tgid`, and that the recording alignments are saved in the correct location. The process itself took approximately three days, but this can still be improved since it was done on a single processor, where multiple could have been used.

4.4 Performance evaluation

It is important to note that due to the fact that there is a systematic delay in Spraaklab's results, alignments will be offset by -50 ms before evaluation. The offset results are identical to the original resuls in all else.

4.4.1 Manual correction of SN sample

As already stated the sample will contain 100 recordings. The trade-off between time spent correcting and the quality of the corrected alignment, due to the author's experience, is not good enough to spend days and days on manually correcting the alignments. The obtained results will still be useful though, by giving a crude indication of the accuracy of the alignments.

In order to prevent bias during the sentence selection process, a random number generator was used to generate sentence indices, the only criterion was the sentence had not already been selected. These hundred sentences were then exported to a Praat friendly format and subsequently opened in Praat. Correcting the boundaries was done on a very basic level, determining by audio and visual wave representation whether the boundary in question was inserted correctly. The corrected version was saved in the same format, and the boundaries could then be processed by looking at the differences between the Praat files, the original version and the corrected version.

Table 10 contains a short summary of the processed sample. There was a total of 59 sentences that did not need correction. Of the sentences that needed to be corrected, all needed one or more boundaries moved or removed, and some sentences were simply recorded incorrectly. Regarding the incorrectly recorded sentences, the transcription in Praat was modified to reflect what was being said in the recording.

Total sentences	100 (%)
Corrected sentences	33 (%)
Incorrectly recorded sentences	8 (%)
Total boundaries	1642
Total wordboundaries	1281 (100%)
Wordboundaries moved	62 (4.84%)
Wordboundaries removed	52 (4.06%)

Table 10: Sample summary

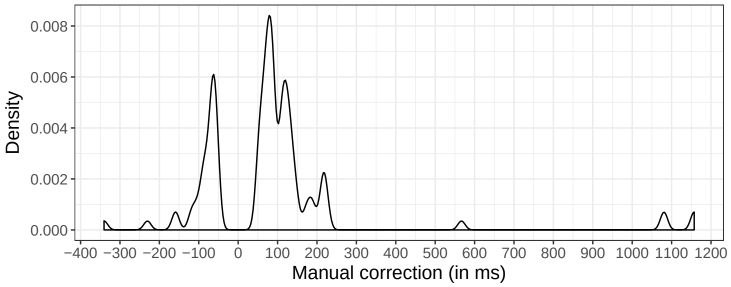


Figure 10: Distribution of manual boundary shifts

Figure 10 contains the distribution for the manual boundary shifts. Excluding some extreme shifts boundaries were not shifted far from their original position. Of all the shifts, 90% were between -102 and 222 ms, which is very reasonably considering that these shifts include the shifts in incorrect recordings.

4.4.2 Duration distribution of frequent words

Different people speaking the same word will result in different word durations, as some people speak slower than others, or have a different way of speaking (putting stress on different phones, or a different accent), however, these durations will still be approximately the same length. The method used here will focus on the durations of frequently spoken words in the SN data and the boundaries of these durations. The outliers will then be checked to see whether the outlier is actually correctly aligned, and if not, whether the recording is at fault or the Spraaklab decoder.

To get a broad evaluation over the word duration distributions it was decided to select words with differing numbers of phones (the phonetic translation provided by the dictionary service), and to pick the most frequently used word with these specific phone lengths. In order to get this information a word frequency list was generated by iterating over all the words in each sentence and keeping a count of every word, all of the capitalised letters were converted to lower case to prevent words appearing twice. Only the sentences that are used to evaluate SN were iterated over, meaning all the sentences for which SN provides a textual representation, 112,054 in total.

The word frequency list was subsequently ordered by frequency and phone length, providing a list of most frequent words with a certain length. The resulting word list can be found in table 11. The phonetic coding used is custom, adding some variations and removing others in order to improve readability of phonetic translations (compared to SAMPA for instance).

Word	(Number of) Phones	Frequency
"de"	(2) [’d’, ’ax’]	94,241
"van"	(3) [’v’, ’a’, ’n’]	26,116
"voor"[a]	(4) [’v’, ’oh’, ’oh’, ’r’]	15,441
"wonen"	(5) [’wv’, ’ow’, ’n’, ’ax’, ’n’]	7,947
"gewild"	(6) [’x’, ’ax’, ’wv’, ’ih’, ’l’, ’t’]	4,305
"kapitein"	(7) [’k’, ’a0’, ’p’, ’iy0’, ’t’, ’ei’, ’n’]	4,344
"tweejarig"	(8) [’t’, ’wv’, ’ey’, ’y’, ’aa’, ’r’, ’ax’, ’x’]	4,039
"verjaardag"	(9) [’v’, ’ax’, ’r’, ’y’, ’aa’, ’r’, ’d’, ’a’, ’x’]	5,495
"onverharde"	(10) [’oh’, ’n’, ’v’, ’ax’, ’r’, ’hh’, ’a’, ’r’, ’d’, ’ax’]	2,035

[a] Usually, "voor" is considered having 3 phones, however, the dictionary service attributes 4 phones to it.

Table 11: Word frequency and number of phones

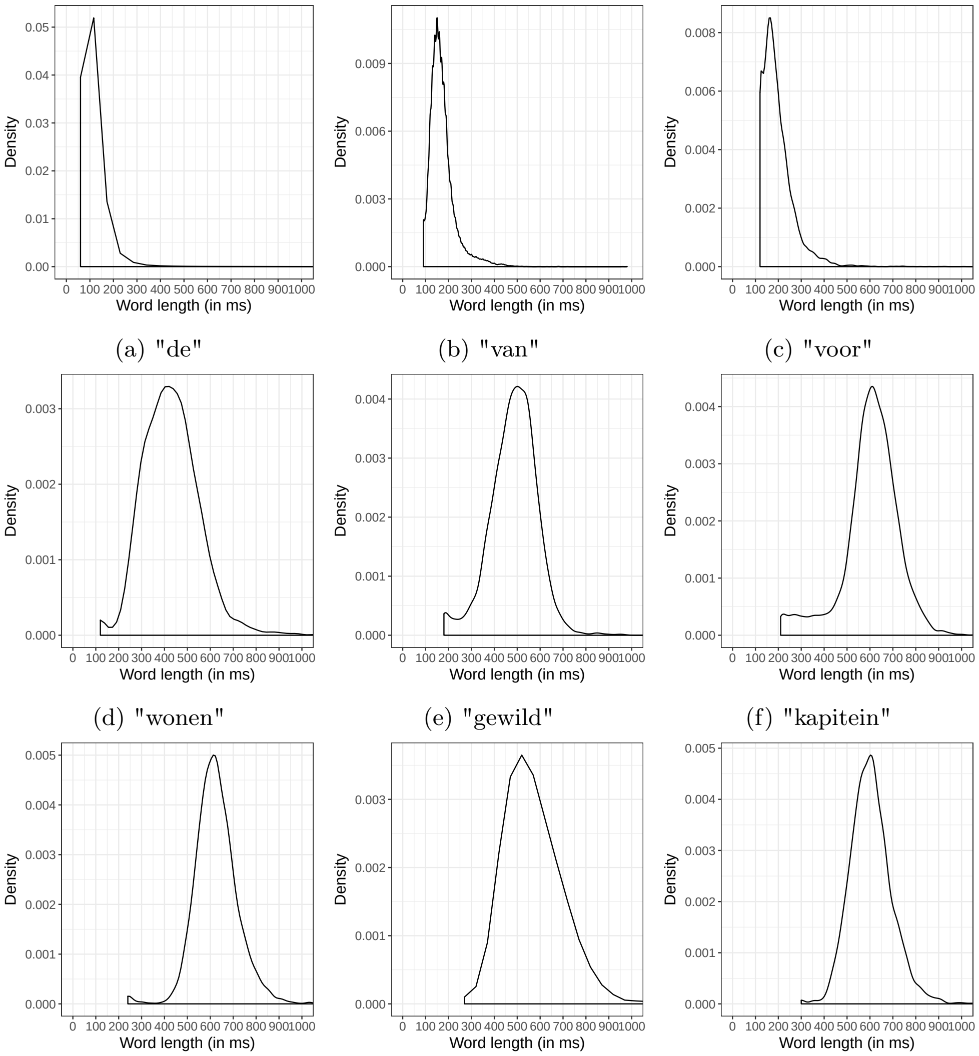
After acquiring this list the data was iterated over again, this time storing all the results on the occurences of the words provided in table 11. Besides the begin and end times of the words, the rid, pid, pgid, and duration were stored. A .csv file was exported for each separate word, in order to be able to analyse the data using R.

The duration distributions for the words mentioned in table 11 can be found in figure 11. For the most part the distributions are not very surprising: the same word spoken by a different person is approximately the same duration, and longer words take more time to say. More interesting are the outliers, these are almost certainly incorrectly aligned, one instance of "de" is a staggering 29 seconds long for example. What is important about these outliers is not that they are incorrect, but why they are incorrect. In the case of an accurate aligner, the reason for an outlier is the recording: white noise, silence, or simply a different sentence. In case of an inaccurate aligner the reason could be anything. Since no aligner is completely perfect, and recordings are always of variably quality, there are two possible sources of errors: the aligner and the recording. Therefore, next to our interest in the overall accuracy, our interest lies in how the aligner reacts when processing a low-quality recording, and if the error source is the aligner, what the possible reason could be.

In order to have a better indication of the quality of the alignments it is important to understand why these outliers are there, which can be done by checking the alignments manually. It is too much work for one person to check each outlier, so the decision was formed to evaluate a selection. Per frequent word, indicated in table 11, 5 alignments will be checked. Two outliers at the short duration end of the distribution, two outliers at the long duration end of the distibution, and one word with the median duration (to check whether the word is correctly aligned at all). The outliers will consist of two extreme outliers located at the 0th and 100th percentile (the minimum and maximum duration availabe) and two that are located at the 5th and 95th percentile, where the word duration is long (or short) but could still reasonably be correct. The instances will be picked semi-randomly, making sure that different recordings have different users to prevent a single user from skewing the results too much.

Table 12 contains a short overview of the outliers checked, the number of correct alignments out of the total alignments checked, and the main reasons for the incorrect alignments should there be any. The total number of alignments is always 9 per percentile, since 9 words are evaluated (the frequented words listed in table 11). Since 'correct' is a rather vague definition, it is defined here using the same 50 ms threshold as earlier: In a correct alignment, both of the word boundaries, begin and end, are within 50 ms of where the manual aligner would have placed them. Incorrect alignments have at least one boundary that is further than 50 ms away from the manually placed counterpart.

As expected, of the manually evaluated alignments, all of the word instances with a median duration were correctly aligned. More interesting are the outliers: when looking at the outliers that are located at the 5th and 95th percentile we see that the alignments are correct most of the time, a good sign, even though more data should be evaluated for a more significant answer. The extreme outliers are all incorrectly aligned, with the exception of one instance of "voor". So far these results are not very surprising either. More interesting is the difference



(a) "de" (b) "van" (c) "voor"

(d) "wonen" (e) "gewild" (f) "kapitein"

(g) "tweejarig" (h) "verjaardag" (i) "onverharde"

Figure 11: Length distributions of frequent words

between misaligned longer outliers, and misaligned shorter outliers. The main errors encountered in the longer outliers are due to background noise, where the decoder assigns a longer portion of the audio to a word. However, in case of the extremely long outliers the length of the recording was also an issue: three out of the nine recordings were spoken aloud just fine, but the recording continued until long after the sentence was finished, messing up the alignment. The shorter alignment mistakes can mainly be attributed to low quality recordings,

Percentile	**Correct / Total**	**Main errors**
P_{100}	0 / 9	Background noise, long recordings
P_{95}	8 / 9	Low quality recording
P_{50}	9 / 9	N/A
P_{5}	7 / 9	Low quality recording, incorrectly spoken
P_{0}	1 / 9	Recording without speech, incorrectly spoken

Table 12: Overview of manually checked distribution samples

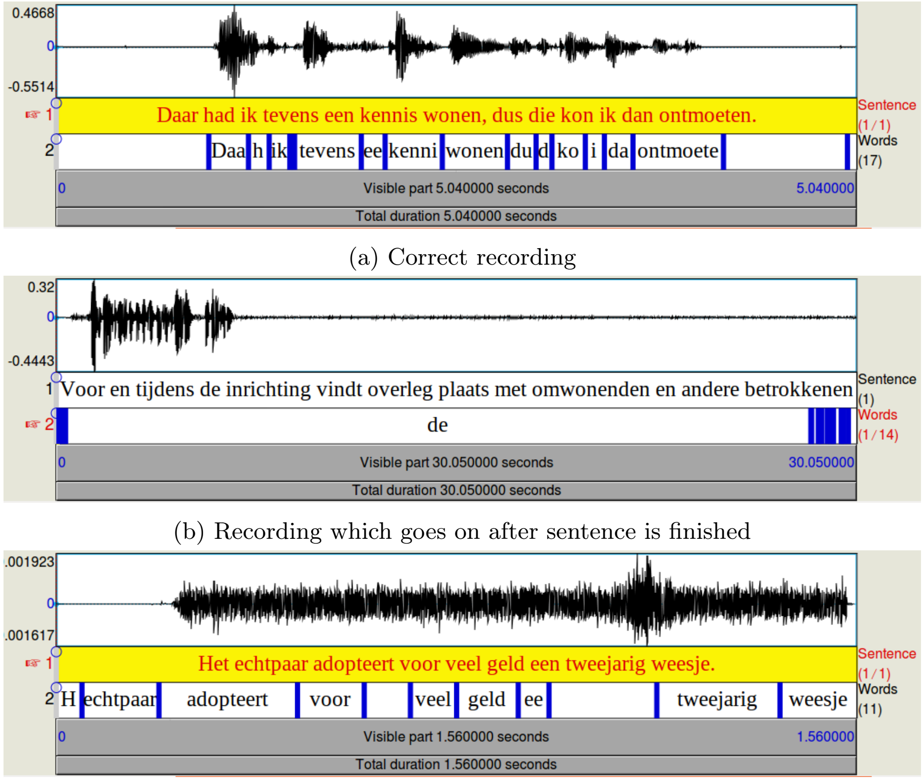
where the range of the pitch is smaller, making the recording sound muffled. Also an issue were incorrectly recorded sentences, adding or leaving out words. With no place to align these words, the entire alignment is thrown out of wack. In case of the extremely short outliers, five of the nine recordings contained no speech (only low volume background noise), causing the decoder to attribute very small portions of audio to the transcription words. Figure 12 contains examples of a correct recording, a too long recording, and an empty recording as seen in Praat.

Of all of these errors, most consist of an error caused by the recording, a problem which is hard for a speech decoder to fix in any case, so this is not very problematic. One type of case should have correctly aligned however, the recordings that went on for a long time after the sentence was finished. The background noise was very small, and the decoder should have been to recognize the silence at the end, correctly aligning the rest of the sentence.

There is one last detail of the distributions in figure 11 that requires attention. All of the distributions have a cutoff at the beginning of the distribution, this is strange because even though the words aligned might be wrong the curve should still be smooth at the beginning given enough data. In order to determine the source, sentences at the 0th percentile were checked for all of the distributions. As visible in the distributions the cutoff grows in accordance with the number of phones a word contains, this was checked and confirmed when looking at the produced alignments on a phonetic level, the phone times produced by Spraaklab. Specifically, the Spraaklab aligner assigns a minimum of 30 ms to each phone in a word, regardless of whether that phone is present in the audio data.

The reason for this is that the Spraaklab aligner uses three states per phone when recognizing the phones, in which the smallest time it can assign to a state is 10 ms (the frame size). Combined with the fact that the aligner needs the phonetic translations, and can only make a word shorter if there is a shorter phonetic translation, this results in the cutoff.

When looking at the cutoff times this becomes obvious as well, all of the cutoff times are 30 ms times the number of phones present in that word. There is one notable exception though, the word "wonen" has a cutoff time of 120 ms, but has 5 phones in table 11, which would result in a cutoff time of 150 ms. An



0.4668
0
-0.5514
1
Daar had ik tevens een kennis wonen, dus die kon ik dan ontmoeten.
2
Daa h ik tevens ee kenni wonen du d ko i da ontmoete
Sentence (1/1)
Words (17)
0
Visible part 5.040000 seconds
5.040000
Total duration 5.040000 seconds
0.32
0
-0.4443
1 Voor en tijdens de inrichting vindt overleg plaats met omwonenden en andere betrokkenen
2
de
Sentence (1)
Words (1/14)
0
Visible part 30.050000 seconds
30.050000
Total duration 30.050000 seconds
.001923
0
.001617
1
Het echtpaar adopteert voor veel geld een tweejarig weesje.
2 H echtpaar adopteert voor veel geld ee tweejarig weesje
Sentence (1/1)
Words (11)
0
Visible part 1.560000 seconds
1.560000
Total duration 1.560000 seconds

(a) Correct recording

(b) Recording which goes on after sentence is finished

(c) Recording without speech

Figure 12: Examples of recordings and their alignments in Praat

inspection of the phones present in the alignment reveals the cause in this case, "wonen" (as well as most other words) has different phonetic translation possibilities in the Spraaklab dictionary service, including a translation consisting of 4 phones[1], resulting in the 120 ms cutoff.

This cutoff could theoretically lead to correctly aligned to be recognized as incorrectly aligned: a word which is actually 80 ms in duration, but has a 140 ms cutoff, and either the begin or end boundary lined up, will have a boundary which is off by 60 ms, which is outside the threshold. However, words with a duration this short hardly (if ever) occur, and therefore it is not something to worry about.

4.4.3 Conclusion

When considering the evaluated data provided in this chapter, Spraaklab performs well. The sample results show that out of 100 sentences 33 had to be

[1] Wonen has the phonetic translation ['wv','ow','n','ax','n'], as well as ['wv','ow','n','ax'].

corrected. It also proved that not all recorded sentences can be aligned correctly due to incorrect recordings, in this case 8 out of 100. This can and should be taken into account when evaluating the rest of the SN data. Also, even though the alignment accuracy on the sample can only be considered a rough indication, it is comparable to the accuracy of Spraaklab on the CGN, meaning that the accuracy of Spraaklab on CGN can probably be taken as a good estimate for Spraaklab's accuracy on unknown corpora.

Looking at the duration distributions of frequently used words, setting aside the problems with the minimal phone durations and the longer recordings, the length distribution data indicates that the Spraaklab speech decoder accurately aligns the individual words, compared to the CGN alignment accuracy, if the sentence is recorded properly and of decent quality. It is possible that boundaries for words with a duration less than its cutoff time could be misaligned due to the minimum time assigned to phones, but an example of this has not yet been encountered. Finally, if a recording contains a long silence at the end the aligner has problems aligning the words correctly. The reason for this could not be found, and no estimate could be made of the amount of recordings in which this is the case.

Overall, the recordings of low quality account for the majority of incorrectly aligned sentences, so filtering these out is important for ease of future research. A possible filter method which could be used is checking aligned word duration, the duration distributions (combined with the manually checked samples) show that extremely long or short word durations indicate that a sentence is probably not aligned correctly. Checking sentence duration could also be a filter for very obvious unusable recordings, but word duration will also filter out unusable sentence alignments that do have a duration which is approximate to the duration of a correct sentence alignment. Calculating these duration limits is outside the scope of this thesis, but it could be interesting for future research.

5 Detecting mistakes using Spraaklab scores

The part of the CGN used in this research is manually verified and corrected, including the transcriptions, so it is known that there are no reading or transcription mistakes. This allows for the Spraaklab alignment accuracy to be tested. The SN data does not provide this certainty, there is audio and textual data present, but users still make recording mistakes, and it is important to be able to identify recordings in which these are made.

Now that we have verified the accuracy of Spraaklab on the CGN and SN, it is possible to take a more detailed look at the word recognition scores that underly the alignments. Spraaklab assigns scores to sections of the audio data corresponding to words in order to find the optimal alignment: the better a word fits the audio, the higher the score. The question is whether these scores will allow us to deduce whether recordings contain a mistake, and should this be the case, what type of mistake it is.

Of course, there are many different ways in which mistakes can be made, which is why we will be focusing on some specific mistakes: words that are differently spoken from what they are supposed to be, words that are added while speaking, and words that are forgotten. The final aim of this chapter is to find an heuristic which uses the word scores and with which it is possible to assign a *correct* or *incorrect* label to an aligned sentence. *Correct* sentences are recorded word-by-word, and *incorrect* sentences have a reading mistake made by the user in them.

In order to find a heuristic, this process will first be simulated using the CGN, by fabricating mistakes. These mistakes will be fabricated by adapting the textual representation of a recording: a changed word mistake can be simulated by substituting one word in the text for a different word; an added spoken word mistake can be simulated by removing a word in the text; and a forgotten word can be simulated by adding a word in the text.

It is important to note again that the CGN data has *correct* and (fabricated) *incorrect* recordings that have been labeled correctly. The SN data does not, although we do know that it contains *incorrect* recordings due to the results in the previous section. The evaluation of the CGN is used to determine word score thresholds with which an algorithm can detect user made mistakes in recordings in SN (or other corpora) *without* having correct recordings to compare them to, in other words, without knowing whether the recording is correct or incorrect.

5.1 Methodology

One reading mistake by the user can affect the word score of adjacent word boundaries, but most of all the score of the mistaken word in question is affected. The question is at what word score threshold the score becomes low enough to be able to label a recording as *incorrect*? In order to find this threshold, this section will focus on the three different types of mistakes mentioned earlier, and each of these needs to be handled slightly different:

- **Word change** Whether a word in the sentence was misread by the user. It will be checked by substituting a word in the text by a word of the same phonetic length.
- **Spoken word addition** Whether a spoken word has been added to the sentence, but is not present in the text. Will be checked by removing a word from a random position in the sentence.
- **Forgotten word** Whether the user forgot to say a word. Will be checked by adding a random word of arbitrary phonetic length to a random position in the text.

By processing both the correct and incorrect versions of the CGN recordings a comparison can be made, and a possible trend between the *correct* and *incorrect* word scores can be found and used to determine a word score threshold for labeling sentences as *incorrect*. The audio data will not be edited besides splitting up large audio files to align the data sentence by sentence, as also carried out in an earlier section. After the CGN data has been processed and evaluated in this way, the SN data will be processed and evaluated.

5.2 Data and processing

Since this part of the research uses the same type of processing as the previous sections, the evaluation part of the process will be carried out on the same part of the CGN used previously. The same SN sample of 100 sentences will be used as well, we know which recordings are *correct* and which are *incorrect*, and this enables us to measure the quality of the heuristic.

The fabricated data is different from previous sections in the way that even though previous sections contain some data editing, the editing performed (e.g., post-processing, or only processing partial audio data) was always defined by the original, raw, data (either originally provided by the CGN, or by the Spraaklab results). The fabricated mistakes are not. We try to make the fabricated mistakes as natural as possible, but this is hard due to the unpredictable nature of mistakes. Evaluating on mistakes that were made by actual users would provide more usable data, something which might be the focus of further research.

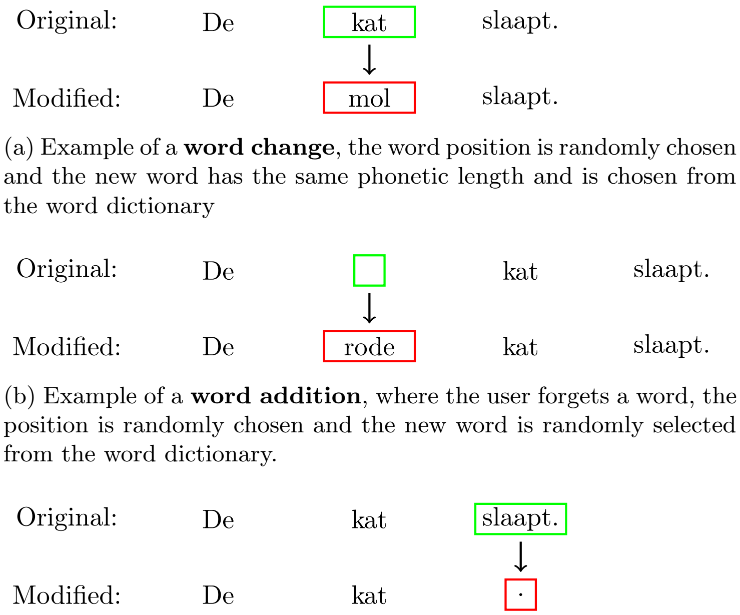
5.2.1 Fabricating the incorrect data

This section will describe how the mistakes will be added to the original data, while trying to cover as many possible different mistakes as possible. The relative position in the sentence at which a mistake is added is decided randomly using a pseudo-random number generator, we make the assumption that this allows for a natural spread in the absence of data which detail these mistakes. Another assumption made is that when a user makes a mistake by either adding or changing a word, the word will be semantically close to the original word.

In order to facilitate this as much as possible a word dictionary has been compiled, containing all the words present in the processed data (the CGN sample of 10 fragments also used in an earlier section). Using words from the CGN sample for selection increases the chance that the word is related to the subject of the recording, compared to simply picking a word from the Dutch language at random.

When changing a word, a word is chosen with the same phonetic length as the original word, under the assumption that when a user changes a word, it will be close in length to the original word. Finally, the word removal mistake is implemented by simply removing the word at the position chosen.

Figure 13 contains examples that cover all types of mistakes that will be manually incorporated, the examples were made by the author, and are not present in the CGN or SN.



(a) Example of a **word change**, the word position is randomly chosen and the new word has the same phonetic length and is chosen from the word dictionary

(b) Example of a **word addition**, where the user forgets a word, the position is randomly chosen and the new word is randomly selected from the word dictionary.

(c) Example of a **word removal**, where the user adds a word. A word is randomly chosen and simply removed.

Figure 13: Examples of the manually added mistakes in the text: word change, word addition, word removal.

All of the mistakes are introduced at the moment before the audio data and sentence enter the aligner and produce a forced alignment. Choosing this point allows the program to very easily generate both the *correct* and *incorrect* results, by running the aligner twice.

5.2.2 The scores

The Spraaklab aligner produces results based on the optimal sentence score, which is the highest possible sentence score. Word scores are assigned to sections

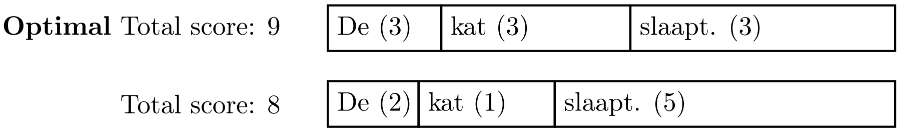


Figure 14: Examples of a scored sentence in which the optimal sentence score does not maximize all word scores. Word scores are enclosed by the parentheses.

of the audio corresponding to words. Different sections will produce different scores for a word, depending on how well the section fits the acoustic model of the word. The sentence score is the most important to the aligner, which means that a word might get a lower score so that other words have relatively higher scores. Figure 14 contains a very simplified example which illustrates how a word score has a lower score than it could have gotten, in order to get a higher sentence score, look at the word 'slaapt' in the two examples. This is important to keep in mind, because this means that a lower score for a word does not mean that the word is necessarily incorrect. The text was created by the author for the example, and is not present in the CGN or SN.

The word recognition scores are located in the Spraaklab results [2]. These scores are calculated by taking the sum of all phone scores belonging to that word, which in turn are calculated by taking the mean of the frame log likelihoods. It describes how well a phone has been recognized in comparison to other phones. In essence, the more positive a word score is, the more confident the aligner is in assigning the word to that section.

5.3 Evaluating the CGN results

In order to determine the word threshold for labeling recordings as *incorrect* it is necessary to know which word scores to compare. This will differ per type of mistake, as an added word has no counterpart in the original sentence to compare its score with, and a removed word has no score itself.

The quality of the heuristic will be measured using false negative (FN) and false positive (FP) rates, the probability that a *correct* recording is labeled as *incorrect* and the probability that an *incorrect* recording is labeled as *correct*, respectively. Considering that future research will obtain better results using *correct* recordings, a lower FP rate has a higher priority than a low FN rate, although the FN rate will also be kept within reason.

5.3.1 The changed spoken word mistake

Figure 15 contains the word score distribution for the original and the changed words. Figure 16 contains the detection error tradeoff (DET) curve based on the score threshold, which shows the FN rate plotted against the FP rate (the x- and y-axes are scaled logarithmically). The DET curve shows that there is

[2] The word scores are represented by the `llh` tag

an equal error rate (EER) of approximately 6.1%, a threshold at which both the FP and FN rate are 6.1%, in this case this point is located at a word score threshold of -0.63. Figure 16 shows that if the FP rate is lowered, the FN rate goes up disproportionately. A 5% FP rate coincides with a 10% FN rate, and a 3% FP rate coincides with a 40% FN rate. In other words, in order to correctly label 97% of the incorrect sentences as *incorrect*, 40% of the *correct* sentences would also be labeled as *incorrect*.

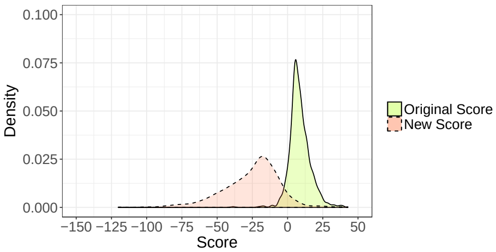


Figure 15: Word score distribution for changed words against original counterparts

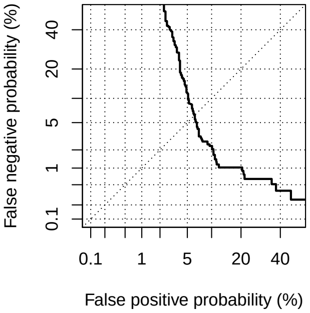


Figure 16: DET curve for labeling correct, and changed, words

SN does not have recordings that are known to be correct, so another method of analysis is also evaluated, which only uses the word scores of the *incorrect* recording. The score of the substituted word is compared to the other words in the sentence, which are still correct.

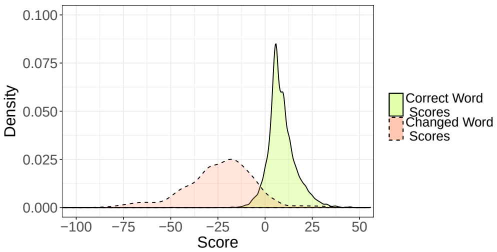


Figure 17: Changed words against correct words in *incorrect* recordings

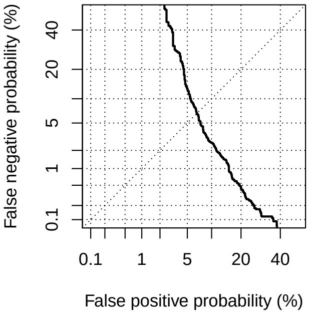


Figure 18: DET curve for changed words

Figures 17 and 18 contain the distribution of the changed word scores and the correct word scores from the incorrect recordings, and the corresponding DET curve, respectively. The EER is 6.6%, very close to the EER between the changed and original word evaluation, and the DET curve is also very similar to the DET curve in figure 16. This means that the effectiveness of the score threshold, when evaluating changed word mistakes, hardly changes, even though only words from the *incorrect* recording are used.

5.3.2 The forgotten word mistake

In case of the mistake where a user forgets to say a word, there is no counterpart word which prevents a direct comparison as a heuristic. However, we know which words are added to the fabricated sentences, and which ones are originally in the text. This allows a comparison between the forgotten word score and the correct word scores. Figure 19 contains the word score distribution, and figure 20 contains the corresponding DET curve. The EER is 7.24%, located at a threshold of appoximately −9.8, and the DET curve shows that a lower FP rate can be accomplished without discarding too many correct recordings.

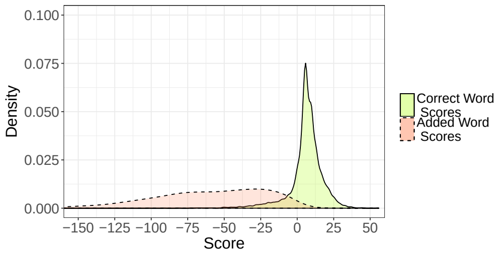


Figure 19: Forgotten word scores against correct word scores

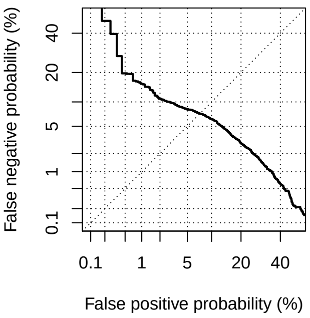
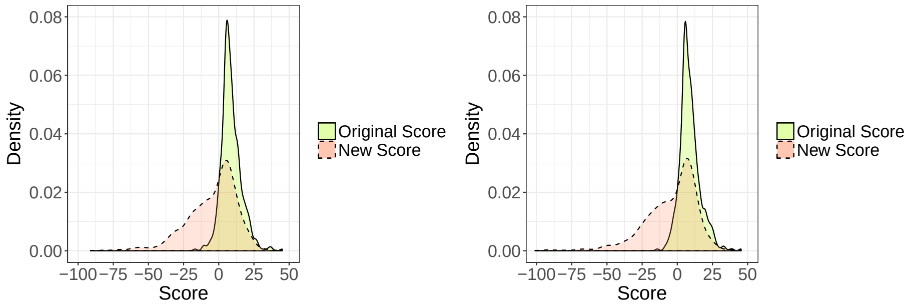


Figure 20: DET curve for added words

Another method to evaluate forgotten word mistakes might be to evaluate the scores of adjacent words. The word present in the text but not in the audio will still be aligned, with a lower word score, meaning that the words preceding and following the forgotten word might also be displaced, resulting in a lower score.

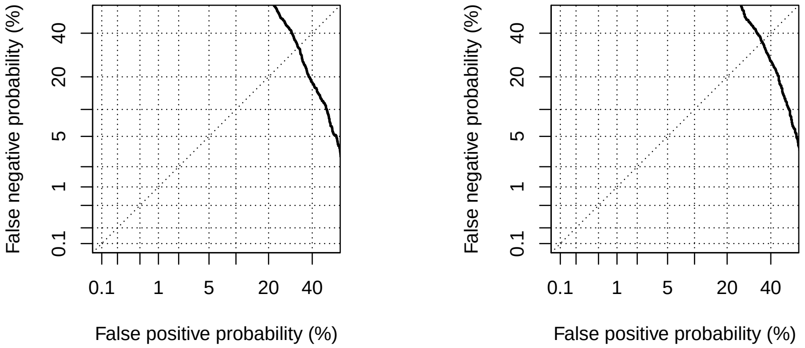
Figure 21 contains the score distributions for the words surrounding the forgotten word. When looking at these it becomes apparent that there is a lot of overlap. The problem that this introduces becomes more obvious in the DET curves belonging to these distribiutions, shown in figure 22. In both cases the EER is close to 40%, and lowering the percentage of FP to less than 10% means



(a) Preceding word scores

(b) Following word scores

Figure 21: Score distributions for words surrounding the forgotten word mistake



(a) Preceding word error tradeoff

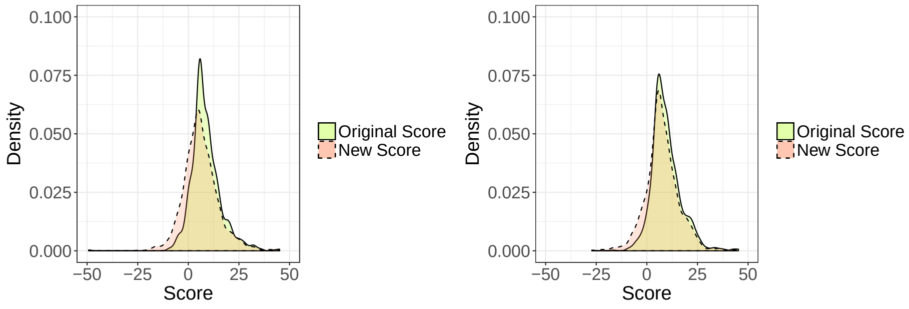
(b) Following word error tradeoff

Figure 22: DET curves for words surrounding the added word mistake

discarding more than 80% of the correct sentences, which is not a satisfactory number. Since the added word scores alone already provide an FP rate with better EER, we will not be using the surrounding words for evaluating forgotten word mistakes.

5.3.3 The added spoken word mistake

When looking at the mistake where a user says a word which is not in the sentence, the only word scores that you can really work with are the scores of the surrounding words, since the spoken word is not in the text, and therefore can not be aligned. Unfortunately, as can be seen in figure 23 these distributions suffer from the same problem as the previous mistake: a large amount of overlap. This makes it an impossible mistake to detect using word scores.



(a) Preceding word scores

(b) Following word scores

Figure 23: Score distributions for words surrounding the added spoken word mistake

However, this only considers the adjacent words as a possible method for detecting a word that was added by the user. Since the aligner regards the additional spoken word as garbage audio, another possibility is that Spraaklab aligner assigns silence to the section containing the additional word, rather than moving the boundaries of the adjacent words, in order to keep the other words in the sentence relatively well aligned. The silence(s) assigned to the added word section would get lower word scores since the section is not actually silent.

These silences can not be compared to the other words since we have no data on how silence scores compare to word scores, but it is possible to compare the added word silences to the other silences in the sentence. The position of the surrounding words is known, so the silences between these two words are considered added word silences, while the other silences in the alignment are considered *correct* silences.

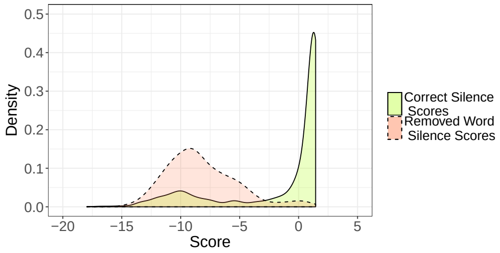


Figure 24: Silence scores assigned to the added words against correct silence scores

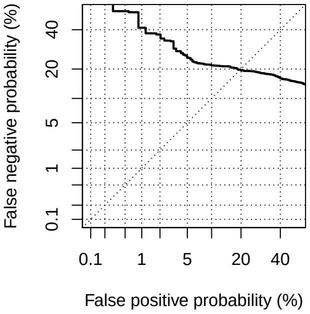


Figure 25: DET curve for added words

Figure 24 contains the silence score distributions for the added word silences and the *correct* silences, figure 25 contains the corresponding DET curve. The EER is 19.31%, which is worse than the previous two mistake types, but better than considering the surrounding words which have an EER close to 50%, practically the same as simply guessing whether a sentence is correct.

The EER is located approximately at a score threshold of −5.8, however, by raising the threshold to −2 it is possible to lower the FP rate by 15% (to 5%), while only raising the FN rate by 5% (to 25%). And since the primary goal is to lower the FP rate, this seems like a reasonable tradeoff.

5.4 Evaluating the mistake results

The goal in evaluating the score distributions and DET curves is to find a heuristic which will allow us to label sentences that are incorrectly recorded by evaluating the word scores in the alignments. One of the assumptions was that surrounding words would be sufficiently impacted that their scores could also be used in supporting the algorithm. Unfortunately, as can be seen in the resulting distributions in this section, this turns out not to be the case.

However, comparing the incorrect word scores to the correct word scores in the sentence showed that there is a discrepancy large enough to largely separate the correct words from the incorrect words in the changed and forgotten word mistakes. And in case of the added spoken word mistake, the silence scores also show promise of being a heuristic by which *incorrect* recordings with this type of mistake can be labeled.

5.4.1 Determining the threshold

All of the mistakes covered in the previous section were shown to have good separation of correct and incorrect recordings when considering their word scores. The question is now how high to set the score threshold which determines whether a recording will be labeled as *incorrect* based on a word score. Silences will be evaluated separately from the word scores, but will also need a score threshold.

A lower threshold will result in a higher FP rate, which means more *correct* recordings will be correctly labeled, but less *incorrect* recordings. Reversely, a higher threshold will result in a higher FN rate, where more *incorrect* recordings will be labeled correctly, but less *correct* recordings. Finally, the threshold can be set so that the FN and FP rate are equal, the EER, but which does require two separate word score evaluations since the EERs for the changed and forgotten word mistake are not equal.

What needs to be considered is how the results gained from the heuristic will be used, which in this case is, among other things, research into dialects. This means that sentences need to be correct in order to get more reliable results for future research. The decision was made to choose a slightly higher threshold, and thus correctly labeling more *incorrect* sentences. This will result in a lower percentage of recordings with user made mistakes still present in the data after filtering out the recordings (either correctly or incorrectly) labeled as *incorrect*.

The changed and forgotten word mistakes will both use the same threshold, since a word score threshold specifically used for changed word mistakes will also identify forgotten word mistakes.

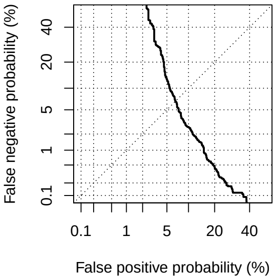


Figure 26: Changed words

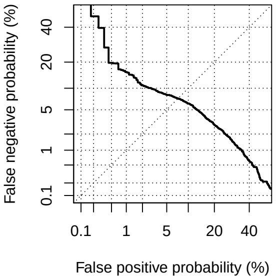


Figure 27: Forgotten words

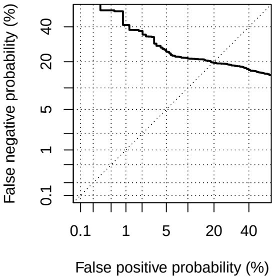


Figure 28: Additional words

Threshold	**FP (Changed)**	**FN (Changed)**	**FP (Forgotten)**	**FN (Forgotten)**
−1	7%	5.8%	1.39%	14.35%
0	6.4%	7.67%	0.69%	16.77%
1	5.5%	10.4%	0.69%	19.4%

Table 13: Word score threshold and corresponding FP and FN rates for the changed word and forgotten word mistakes.

Threshold	**FP (Added)**	**FN (Added)**
−6	21.28%	19.33%
−5	15.16%	21.16%
−4	8.75%	22%
−3	5.83%	23.31%
−2	4.96%	25.14%
−1	4.08%	28.28%
0	2.92%	33.81%
1	0.87%	50.32%

Table 14: Silence score threshold and corresponding FP and FN rates for the added spoken word mistake.

Figures 26, 27, and 28 contain the same DET curves shown in the previous section. Table 13 contains a few word score thresholds and their corresponding FP and FN rates for the first two types of mistakes: changed words and forgotten words. The thresholds are located close to 0 since this gives the best FP rate, while still having a relatively low FN rate. Table 14 contains the silence thresholds and the corresponding FP and FN rates for the added spoken word mistake.

From the tables it becomes clear that the same threshold can not be used for silences and words, a good threshold can be set for both however. Since the focus is more on labeling as many *incorrect* recordings as possible to make future research easier the decision was made to set the word threshold at 0, giving an FP rate of 6.4% for changed word mistakes, and 0.69% for forgotten word mistakes, while also keeping the FN rate relatively low. The silence score threshold will be set at -2, which will discard a quarter of *correct* recordings, but the FP rate will be approximately 5%.

5.5 Evaluating the labeled SN recordings

In order to correctly evaluate the word and silence thresholds obtained in the previous section it is necessary to evaluate a sample from SN in which the correct and incorrect recordings are already known so that the FN and FP rates can be compared. The SN data itself does not provide this information, and therefore a manually evaluated sample is needed. The same sample evaluated in table 10 will also be evaluated here, because it provides the information needed on incorrectly recorded sentences.

Since the method uses two different heuristics, a word score threshold, and a silence score threshold, the change word mistake and the forgotten word mistake will both be treated as a 'word' mistake, while the added spoken word mistake will be treated as a 'silence' mistake. The *incorrect* recordings sample is very small, 8 recordings, but might still provide insights into to the effectiveness of the thresholds when paired with manual evaluation of the mistakes.

Note that we define recordings as *incorrect* due to mistakes made by the user. Some recordings will get scores lower than than the thresholds due to misalignment, however, and therefore be labeled as *incorrect* instead of *correct*. These recordings will be regarded as FN, since the goal is to identify user made mistakes.

5.5.1 The results

The sample contains 100 recordings. Of these recordings, 92 are correctly spoken and 8 are incorrectly spoken. For each recording, all of the words were evaluated against the word threshold, and all of the silences were evaluated against the silence threshold. Table 15 contains the results.

As can be seen, all of the recordings in the SN sample were labeled as *incorrect*. The question now is whether this is the case because the thresholds simply do not work for the SN corpus, or whether this is a calibration problem due to overall SN scores being lower. It is probably the latter, since the acoustic model used by Spraaklab is trained on the CGN, meaning that the words spoken in the CGN will probably fit the model better and therefore obtain higher scores.

Looking at the distributions of the CGN and SN scores, using the mistake evaluation samples, in figure 29 shows that this is indeed the case. The mean

	Labeled as *correct* (%)	**Labeled as *incorrect* (%)**
***Correctly* spoken recordings**	0 (0%)	92 (100%)
***Incorrectly* spoken recordings**	0 (0%)	8 (100%)

Table 15: Amount *correct* and *incorrect* recordings labeled correctly or incorrectly.

score of SN words is 0.8, while the mean score of CGN words is 8.8, a difference large enough to label all recordings as *incorrect* in this SN sample.

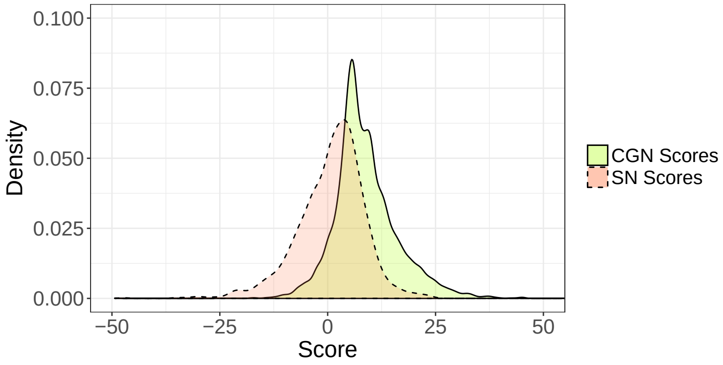


Figure 29: Word score distributions for the CGN sample (10 fragments) and the SN sample (100 sentences)

The mean number of mistaken words labeled per recording, including the *incorrect* recordings, is 5. This raises the question whether the mistaken words in the *incorrect* recordings were actually lower than the correct word scores.

Figure 30 shows the score distributions for mistakes and correct words beneath the threshold. A slight separation between the two distributions can be seen. Unfortunately, there is too little data here to be able to conclude anything, but it does indicate that using a threshold heuristic could still work, given better score calibration.

The mistakes in this SN sample consisted of 4 changed word mistakes, and 4 forgotten word mistakes. Unfortunately there were no added spoken word mistakes to compare to, but figure 31 shows that the correct silences from the CGN and SN have a lot of overlap. There is no incorrect silence data from SN in this sample to confirm this, but the silence threshold might be better calibrated to detect added spoken word mistakes in the SN data than the word threshold is to detect the other mistakes.

In order to get an indication whether the results were more due to the calibration than to the heuristics being ineffective the SN sample was processed two more times using a word thresholds of -8 (the difference between the mean

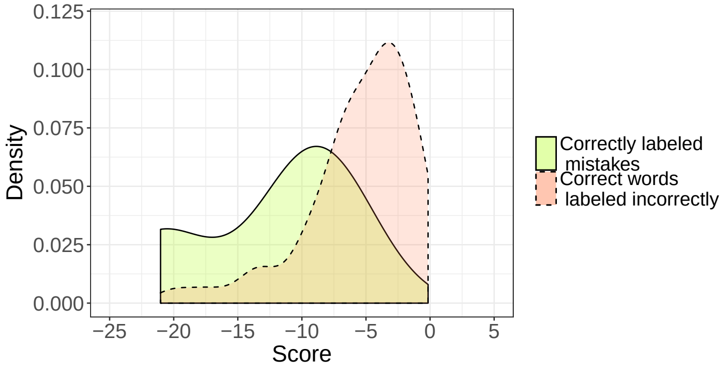


Figure 30: Score distributions for correct words labeled as *incorrect*, and actual mistakes labeled as *incorrect*.

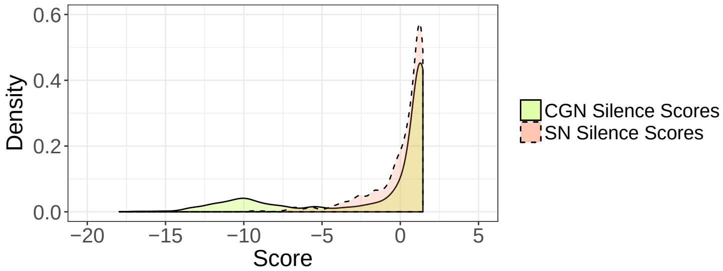


Figure 31: Score distributions for correct CGN silences and correct SN silences.

	Labeled as *correct* (%)	**Labeled as *incorrect* (%)**
***Correctly* spoken (threshold: −8)**	16 (1̃7.4%)	64 (8̃2.6%)
***Incorrectly* spoken (threshold: −8)**	1 (12.5%)	7 (87.5%)
***Correctly* spoken (threshold: −10)**	28 (3̃0.4%)	64 (6̃9.6%)
***Incorrectly* spoken (threshold: −10)**	1 (12.5%)	7 (87.5%)

Table 16: Amount *correct* and *incorrect* recordings labeled correctly or incorrectly using a word threshold of −8 and −10.

CGN and SN word scores) and −10 (to look at the effect of lowering the threshold even more). The results this time, shown in table 16, were more promising: an FP rate of 12.5% and a FN rate of 82.6% for a word score threshold of −8, and an FP rate of 12.5% and a FN rate of 69.9% for a word threshold of −10. This shows that a lower word threshold results in a higher FN and lower FP rate for SN, as it also does for the CGN.

The data in table 16 is not enough to conclude anything, but it shows that heuristics are not necessarily ineffective on the SN data. More manually checked SN data is necessary to form a better conclusion.

5.5.2 Conclusion

The data present in the results is not enough to be able to conclude whether the heuristics work well for labeling incorrectly recorded sentences. However, it is important to note that the does *not* show the heuristics to be ineffective towards this goal. It is probable that the first results, using a word threshold of 0, are due to bad calibration since the SN has lower average word scores than the CGN. This argument is further strengthened by crude followup iterations through the data, showing improvements of the FN rate. However, the results are not strong enough to conclude this with certainty.

The distributions between CGN and SN silences show that these scores have a lot of overlap, but silence is the absence of speech, so it stands to reason that the CGN trained acoustic model assigns approximately equal scores to silences in the CGN and SN, because there is no speech that can be fitted. However, this might be different for recordings with a high level of background noise.

In order to reach a better conclusion more manually verified SN alignments are needed to compare the CGN FP and FN rates to, which could be a subject for future research.

6 Conclusion

The research in this paper was done in order to reach two objectives: To align the currently available data in the Sprekend Nederland corpus, and to develop a method to detect speaker made mistakes in the recordings that are aligned. In order to reach these objectives Spraaklab's alignment accuracy first had to be evaluated in order to have a benchmark to compare the SN alignment accuracy to.

In order to evaluate Spraaklab's alignment accuracy, the Corpus Gesproken Nederlands was used. It was discovered that Spraaklab has a systematic delay of approximately 50 ms in all of its alignments compared to the gold standard. Spraaklab also adds approximately 50 ms to the alignment duration compared to its audio component, and CGN counterpart. Neither the systematic delay nor the elongated duration could be explained, but the elongated duration might have to with the delay, as they are practically of the same duration. It was also determined that neither the silence parameter nor extending the audio with silence at the start and end of the sentence affects the word boundary alignments by more than 10 ms. After compensating for the delay, Spraaklab aligns 90% of the CGN word boundaries within 50 ms of the corresponding boundary in the gold standard, not including silences, where Spraaklab inserts about twice as many, but also shorter in duration.

In order to evaluate the SN alignments, the author manually verfied an SN sample set consisting of 100 sentences. Boundaries off by more than 50 ms were manually corrected. More than 95% of the word boundaries were within the 50 ms threshold, surpassing the benchmark by 5%. Besides alignment accuracy, methods using word and sentence duration for evaluating the alignment quality were proposed for possible future research. Alignments were calculated for all eligible Spraaklab recordings, including the 50 ms delay to keep the data authentic.

Knowing that Spraaklab is capable of accurate alignments, a CGN benchmark was produced for three types of mistakes: changed words, forgotten words and added spoken words. Using a word score threshold, the method produced a 5% false negative rate, against a 10% false positive rate. In order to label added spoken word mistakes, a heuristic was developed using silence scores inserted at that position in the sentence, this heuristic produced a 5% FP rate on the CGN.

Both the word and the silence heuristic were evaluated by labeling the earlier verified SN sample. A word score threshold of 0 and silence score threshold of -2 produced only recordings labeled as incorrect, due to SN scores being lower than CGN scores. These results might improve after calibrating the heuristic against the SN scores. A quick reiteration over the data using a lower threshold shows an improvement in the FN rate. More research, using manually evaluated SN alignments, is needed to obtain better conclusions.

In conclusion, the Spraaklab aligner proves to be 90% accurate to within 50 ms on the CGN, and manual evaluation roughly indicates that it is 95% accurate to within 50 ms on SN. These results are comparable to the results produced in other papers. Compared to Chen et al., Spraaklab performs better when considering the mean boundary shift. Chen et al. produce a mean WBBS of 24.11 ms and a mean WEBS of 29.95 with their best performing aligner, while Spraaklab produces a mean WBBS of 0.18 ms and a mean WEBS of 11.87 ms (after removing the delay). Spraaklab performs only slightly worse when considering the standard deviation. Chen et al. produce a WBBS SD of 27.56 ms and a WEBS SD of 31.68 ms, while Spraaklab produces a WBBS SD of 43.34 ms, and a WEBS SD of 33.12 ms [10]. As planned, all the eligible SN recordings have been aligned and delivered, albeit including the 50 ms delay.

And finally, a method for labeling recordings was shown to be possible on CGN data, and, compared to results produced in other papers, performs well. Srikanth et al. produce an FP rate of 20% and an FN rate of 45.25% on evaluated mispronounced words (non-native spoken English) [15], while Spraaklab produces an FP rate of 6.47% and an FN rate of 7.67% on changed words (the type of mistake used in this research to simulate mispronounced or misread words). Both aligners would need to align the same data for a better comparison. There is no data to compare forgotten and added spoken word FP and FN rates.

Unfortunately, due to a lack of manually verified SN data, Spraaklab could not be shown to be effective on SN data, producing only recordings labeled as *incorrect*. However, there are indications that this is due to threshold calibration: the SN word scores are lower than the CGN word scores on average, and iterations using lower score thresholds showed improvement for the FP rate. Due to the score differences, CGN scores are not usable in determining an effective SN word score threshold. However, the method of using word score and silence score thresholds can probably be shown to be effective on the SN data in further research, by manually evaluating a larger sample and using the results to determine the thresholds.

List of Figures

List of Tables

References

[1] K. Harrenstien. Automatic captions in youtube. `https://googleblog.blogspot.nl/2009/11/automatic-captions-in-youtube.html`, 2009. Accessed: 2017-01-23.

[2] NTR. Ntr maakt blauwdruk van het gesproken nederlands. `http://www.ntr.nl/site/nieuws/NTR-maakt-blauwdruk-van-het-gesproken-Nederlands/134`. Accessed: 2017-01-23.

[3] NTR. App sprekend nederland nadert lancering. `http://www.ntr.nl/site/nieuws/App-Sprekend-Nederland-nadert-lancering/111/`. Accessed: 2017-01-23.

[4] NovoLanguage. `http://www.novolanguage.com/`. Accessed: 2017-01-26.

[5] M. A. H. Huijbregts. *Segmentation, Diarization and Speech Transcription: Surprise Data Unraveled.* PhD thesis, University of Twente, Enschede, November 2008.

[6] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Nagendra Goel, Mirko Hannemann, Yanmin Qian, Petr Schwarz, and Georg Stemmer. The kaldi speech recognition toolkit. In *In IEEE 2011 workshop*, 2011.

[7] Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88, 2002.

[8] Nederlandse Taalunie. Het corpus gesproken nederlands. `http://lands.let.ru.nl/cgn/`. Accessed: 2017-01-26.

[9] Nederlandse Taalunie. Cgn-publicaties. `http://lands.let.ru.nl/cgn/doc_Dutch/topics/project/publ.htm`.
Accessed: 2017-01-26.

[10] Lei Chen, Yang Liu, Mary Harper, Eduardo Maia, and Susan Mcroy. Evaluating factors impacting the accuracy of forced alignments in a multimodal corpus. In *Proc. of Language Resource and Evaluation Conference*, 2004.

[11] Jan Strunk, Florian Schiel, and Frank Seifart. Untrained forced alignment of transcriptions and audio for language documentation corpora using webmaus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*, 2014.

[12] G Saha, Sandipan Chakroborty, and Suman Senapati. A new silence removal and endpoint detection algorithm for speech and speaker recognition applications. In *Proceedings of the 11th national conference on communications (NCC)*, pages 291–295, 2005.

[13] Javier Ramirez, Juan Manuel Górriz, and José Carlos Segura. *Voice activity detection. fundamentals and speech recognition system robustness.* INTECH Open Access Publisher NewYork, 2007.

[14] Nederlandse Taalunie. The .skp format. `http://lands.let.ru.nl/cgn/doc_English/topics/version_1.0/formats/xml/skp.htm`. Accessed: 2017-01-27.

[15] Ronanki Srikanth, Li Bo, and James Salsman. Automatic pronunciation evaluation and mispronunciation detection using cmusphinx. In *Google Summer of Code, CMU Sphinx*, 2012.