Utrecht University

Master's Thesis
Linguistics: The Study of the Language Faculty

The "Sprekend Nederland" project applied to accent location

Georg Lohfink

Supervisors: Gerrit Bloothooft & David van Leeuwen

June 2017

Contents

1 Introduction **4**

2 Data **5**
- 2.1 Recordings . . . 6
- 2.2 Meta-Data . . . 6
  - 2.2.1 Location . . . 7
  - 2.2.2 Age . . . 8
  - 2.2.3 Gender . . . 9
  - 2.2.4 Other Nationalities or Ethnicities . . . 9
  - 2.2.5 Education and Work . . . 9
- 2.3 Judgements about Fellow Participants . . . 10
  - 2.3.1 Perceived Accentedness . . . 10
  - 2.3.2 Location Judgements . . . 11
- 2.4 Challenges for Accent Location . . . 12

3 Data Preparation **13**
- 3.1 Data Selection . . . 13
- 3.2 Feature Extraction . . . 13
- 3.3 Principal Component Analysis . . . 14
- 3.4 Partitioning for Cross-Validation . . . 15
- 3.5 Storing . . . 17

4 Accent Location as a Classification Problem **17**
- 4.1 A Model for Classification . . . 18
  - 4.1.1 Neural Networks . . . 18
  - 4.1.2 Alternatives to Neural Networks . . . 18
- 4.2 Implementation . . . 19
  - 4.2.1 Main Script . . . 19
  - 4.2.2 Dataset Class . . . 21
  - 4.2.3 Hardware . . . 22
- 4.3 Optimisation . . . 22
  - 4.3.1 Learning Rate and Learning Rate Decay . . . 23
  - 4.3.2 Training with Mini-Batches . . . 24
  - 4.3.3 Dropout . . . 26
  - 4.3.4 Prior Class Probabilities . . . 27
  - 4.3.5 Training with Selections of Data . . . 31
- 4.4 Alternative Classes . . . 33
  - 4.4.1 Clustering: Finding Divisions in the Population Distribution . . . 33

4.4.2 Repeated Clustering 36
4.4.3 Manually Corrected Clusters 37
4.5 Results 38

5 Accent Location as a Regression Problem 39
5.1 Reinterpretation of Classification Model's Output 40
5.2 A Regression Neural Network for Predicting Coordinates . . . 41
5.3 Optimisation 41
5.4 Results 43

6 Comparison with Human Performance 46
6.1 Human versus Computer Perception of Accents 46
6.2 Classification 48
6.3 Regression 50

7 Discussion 53

8 Conclusion 55

References 57

Abstract

In this thesis, two versions of an accent location system are introduced which have been built using data from the *Sprekend Nederland* project. These include short recordings of participants' speech, judgements about other participants' speech and meta-data such as the participants' locations. This thesis aims at exploring the questions of how an accent location system can be implemented with the given data and what performance can eventually be achieved.

Both versions of the system were implemented as feed-forward neural networks taking i-vectors as their input which had previously been extracted from the recordings. The first version was built to classify recordings as corresponding to one out of twelve accent regions whereas the second version predicted coordinates of the speakers' locations.

Major challenges faced during engineering the system were due to dealing with an unbalanced dataset which was especially dominated by young participants living in larger cities such as Rotterdam, Amsterdam and Utrecht. Furthermore, participants' self-reporting, judgements of fellow participants, and the results of a principal component analysis of the used features, all indicated that the recordings would contain just little speech with local accents. Building an accent location system on these data consequently required exploiting scarce cues of these accents.

Both versions of the system showed a performance which was just little above chance level. However, human listeners who had been asked to guess participants' locations based on the same recordings could overall not outperform the system. While acknowledging further potential improvements, this led to the conclusion that most of the available cues of local accentedness must have been exploited during the system's training.

1 Introduction

Towards the end of 2015, the project *Sprekend Nederland* was launched by the Dutch broadcast organisation NTR, the Netherlands Organisation for Scientific Research (NWO) and a number of researchers from various Dutch universities. Its aim is to study how Dutch is currently spoken across the Netherlands as well as people's attitudes towards speakers of its various accents and dialects. The project's main tool is a smart phone application via which data from participants are collected. These include recordings of read aloud and spontaneous speech, judgements and different meta-data such as age, education, gender and location.

Van Leeuwen and Orr (2016) outline the *Sprekend Nederland's* potential application to accent location. This task is described as finding a speaker's origin location, given a speech sample. The aim of the thesis work was to find ways of implementing an accent location system using the project's data. In doing so, the focus was explicitly on local accents in the sense of variation on the phonetic and phonological level. Additional aspects commonly covered by the term "dialect" such as lexical differences were not considered. The two main questions of this thesis are how an accent location system can be implemented with the given data and what performance can eventually be achieved.

In recent years, extensive work has been carried out on the task of accent recognition. Gaining knowledge about a speaker's accent can be advantageous if, for instance, applied to speech recognition. Systems accounting for such variability show lower recognition errors (e.g., Biadsy (2011)). Furthermore, accent recognition is useful in speaker identification and, therefore, also has forensic applications (Jessen, 2007). The task of recognising an accent is often times defined as a classification problem. That is, systems trained on a set of accents aim to find the most suitable accent label for a given recording. For such a system to work well, the different accents need to be distinct. However, if there are similarities and overlaps among some of the accents in question, systems show higher confusion rates, i.e., recordings are more often misclassified as a similar accent. Given the focus is on identifying foreign accents (e.g., Arslan and Hansen (1996); Bahari, Saeidi, and van Leeuwen (2013)) this is less of a problem. However, separability is less guaranteed if one aims to classify different local accents. For instance, in Hanani's (2012) dissertation, which focuses on human and automatic recognition of British accents, humans are reported to be more likely to confuse geographically close accents. Partially, the same problem is encountered in automatic recognition.

Commonly, dialects and local accents of a language are described by defin-

ing a set of separate regions. For Dutch, dialects have, for instance, been mapped by Daan and Blok (1969). However, local differences in how Dutch is pronounced across the Netherlands can also be described as a continuum (e.g., Heeringa (2004)).

The question of how an accent location system should be implemented is addressed in two different ways in this thesis. First, the task is defined as a classification problem. As such, it is not different from common applications of accent recognition. Below, models are introduced which were built to classify speakers as belonging to one out of twelve accents regions. The second approach to accent location treats local accents as a continuous phenomenon and defines it as a regression problem. The corresponding models make predictions about a speaker's location in terms of coordinates. Both the classification and the regression approach are discussed by van Leeuwen and Orr (2016). However, the implementations presented below abstract the accent location task by not accounting for the participants' location histories nor their age at the times of living in different places.

In preparation for building an accent location system, it was necessary to first analyse the available data, select useful parts and extract features from audio material. These steps are covered in section 2 and 3. Of course, the implementation of a system for accent location involved making a number of technical decisions. After having experimented with several different machine learning algorithms, the final models were all implemented as artificial neural networks. These types of models are potentially very powerful, however, tailoring these to the specific task of accent location proved challenging. Hence, this thesis includes detailed descriptions of the difficulties encountered in implementing the system. Section 4 deals with creating models for accent classification. In section 5, changes are introduced converting the existing models to regression models which predict speakers' locations in terms of coordinates. For a subset of the available recordings, the speakers' locations have been guessed by human listeners. In section 6, their performance is compared to the models' predictions. This thesis is concluded by a discussion (section 7) highlighting the system's shortcomings and potential ways of improvement.

2 Data

In this section, a detailed account of the available data is presented. These consist of recordings, meta-data and judgements. An outlook will be given about the challenges these data pose to creating models for accent location.

The data used for this thesis only include recordings which had been col-

lected by March 2016 and corresponding meta-data and judgements which were last updated in August of the same year. However, the project continued recruiting new participants and its smart phone application stayed activated allowing for more recordings and judgements to be collected. For the task of building a system for accent location, it was necessary to make a selection of useful data. Only participants who had submitted a recording of at least ten seconds and who had also indicated their location by corresponding coordinates were considered. Less than half of the people who had signed up via the application had completed the tasks to that degree. If not indicated otherwise, "participants" shall further strictly just refer to those 2191 people whose data were used.

2.1 Recordings

The application includes tasks which prompt the participants to make recordings of their own speech. These tasks include reading out word lists and sentences as well as picture naming and giving detailed descriptions in own words. Since recordings were required to have a minimum length of ten seconds the selected ones contain especially spontaneous speech data. In total, these amounted to 13781. The recordings were made using the participant's own devices. Hence a variety of different microphones were used. On manual inspection, the overall quality of the recordings was judged to be good.

2.2 Meta-Data

Although the application includes a questionnaire including 41 questions which would help to locate the participant's accents and even allow to make statements about their potential sociolects and ethnolects, only few answered all of them. Responses are missing to such a degree that one would have to exclude most participants if one wanted to make use of just the data which are accompanied by mostly complete questionnaires. The application had been designed such that the order of questions was not randomised. Consequently, the later a question was asked the fewer participants gave an answer. Figure 1 illustrates this drop in response rates.

Van Leeuwen and Orr (2016) have previously presented some early statistics about participation. The meta-data regarding location, gender and age which are presented below are more recent, however, barely differ.

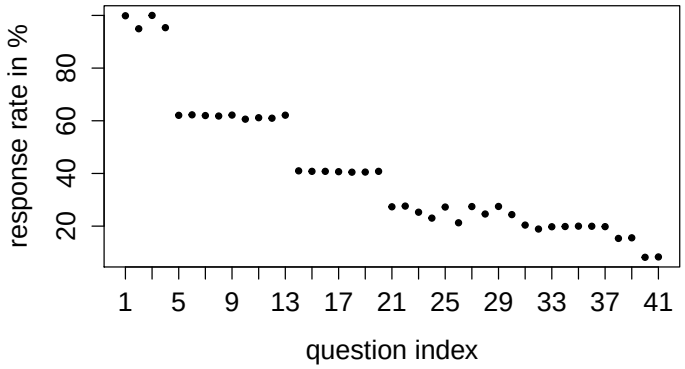


Figure 1: Response rates per question

2.2.1 Location

For this project, coordinates were used which participants directly provided by clicking on the corresponding location on a map. Out of three questions involving this task, one required the participants to specify a location corresponding to their own accent.[1] The resulting coordinates were used where available. However, just a small number of participants gave an answer to this question due to the fact that it appeared as the twenty-first in the application (see Figure 1). All other coordinates were taken from answers to the question asking where within the Netherlands they have lived the longest time.[2] These locations were assumed to correspond to the areas which mostly shaped the participants' accents. Of course, this will not always be the case. For a person to adapt to a certain regional accent will, for instance, also depend on the age at the time of moving to the according region (Chambers, 1992). Responses to a third question asking "Where are you from?"[3] were not used since its meaning is somewhat ambiguous. People answer this question depending on context (Myers, 2006). It may be read to inquire about a person's birthplace, or where she or he currently lives. In real life, a person may respond to this question by specifying both. However, this option was not available in the questionnaire.

The participants' coordinates can be used to match them with a province or any other geographical region. Comparing the data to the official popu-

[1] "Waar plaats je je eigen accent op de kaart?"

[2] "Waar heb je de meeste jaren binnen Nederland gewoond?"

[3] "Waar kom je vandaan?"

lation statistics (Centraal Bureau voor de Statistiek, 2016c), there are more participants from provinces with a higher population density. As Figure 2 illustrates, the number of participants per province and the provinces' numbers of inhabitants are strongly correlated $r(10) = .97$, $p < .001$. In this

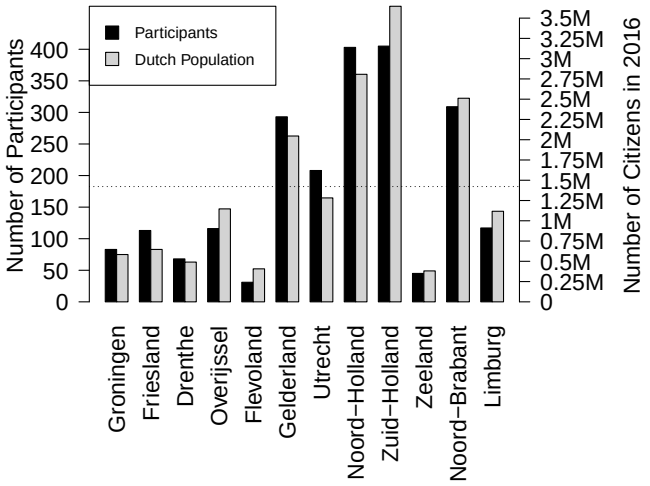


Figure 2: Distribution of participants per province compared to the number of citizens per province in 2016. Bars have been scaled to have the same mean length for both participants and population. The dotted line indicates the average value for both categories.

work, provinces will be referred to purely for illustrative reasons. However, other classes which are potentially more meaningful for the purpose of accent location will also be discussed.

2.2.2 Age

62% of the participants also indicated their year of birth. As Figure 3 shows, participants were mostly younger people in their twenties. A direct comparison with the Dutch age distribution (Centraal Bureau voor de Statistiek, 2016a) shows that age groups are not well represented by the dataset. There was a comparably weaker correlation between participants and citizens per age group, $r(61) = .52$, $p < .001$.

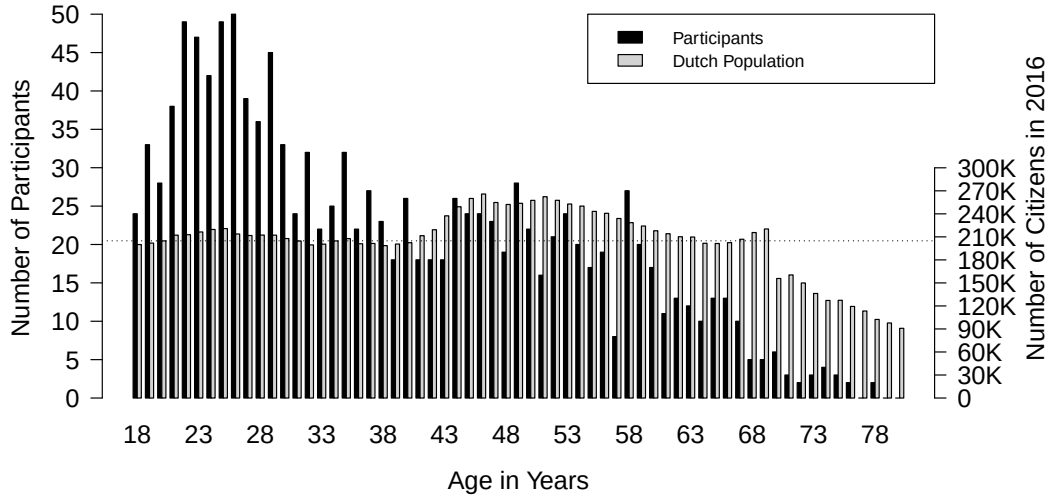


Figure 3: Age distribution of participants compared to the Dutch age distribution in 2016. Bars have been scaled to have the same mean length for both participants and population. The dotted line indicates the average value for both categories.

2.2.3 Gender

61% of the participants indicated their gender. Of those who did, 56.04% were female, 43.74% were male and 0.22% identified as neither of these two. Females are somewhat over-represented in comparison to the Dutch population, of which 50.4% were recorded as females by Statistics Netherlands in 2016 (Centraal Bureau voor de Statistiek, 2016a).

2.2.4 Other Nationalities or Ethnicities

Participants were also asked whether next to being Dutch they would count themselves to one or more other nationalities or ethnicities. However, only 25% gave an answer. Of those, 77% indicated that they consider themselves as only Dutch. The remaining participants are spread across 16 groups. None of these groups count more than 10 participants with the exception of Western Europe and Dutch East Indies (38 and 21 participants respectively).

2.2.5 Education and Work

With a response rate of 20% , even fewer participants stated their highest level of education. Of those, exactly two thirds had enjoyed higher education

at a university or a vocational university. If this is also true for those who did not respond, this group is by far over-represented. According to Statistics Netherlands (Centraal Bureau voor de Statistiek, 2016b), only about one third of the Dutch held a corresponding degree in 2015. When asked about their work, 57% (of those 20% who gave an answer) stated a profession which usually requires higher education, e.g., doctor, scientist, manager or teacher.

2.3 Judgements about Fellow Participants

A main focus of the Sprekend Nederland project is on how people are judged based on how they speak. Its smart phone application includes a large number of tasks prompting participants to make judgements about others who already have submitted a recording of their own speech. Most of these judgements have no relevance for this work as these concern aspects such as attractiveness, intelligence, appeal and trustworthiness. However, they also include ratings of accentedness as well as location judgements. The former were attempted to be applied in training models and the latter were used in evaluating the models' performance.

2.3.1 Perceived Accentedness

As part of the questionnaire, the participants were directly asked whether they speak a Dutch dialect. Of all participants, 20% replied with 'yes', 41% with 'no' and the remaining 39% did not answer the question. Among the tasks involving judgements about other speakers, participants had to tell whether they consider another participant to have a strong accent after having listened to that person's recording. They could supply their answers via a discrete slider with values ranging from 1 to 7 indicating strongest disagreement and agreement respectively. Initially, the slider was positioned in the centre at value 4. Of the participants, 95% received at least one such rating. On average, most participants were rated to rather not have a strong accent whereby the judging participants mostly avoided extreme values on both ends of the scale (see Figure 4).

These accentedness ratings appear to be incongruous with the percentage of participants mentioned above who stated that they do not speak a dialect. One may expect that their ratings should be accordingly low. The absence of a Dutch dialect, however, does not guarantee that their speech was not accented in other ways. In fact, this group was still rated with 3.4 on average. It is possible that speakers' self-assessments were not always correct or that the participants who did the rating did not always use the slider as intended. Selecting the central value 4 may have been the choice for some who wanted

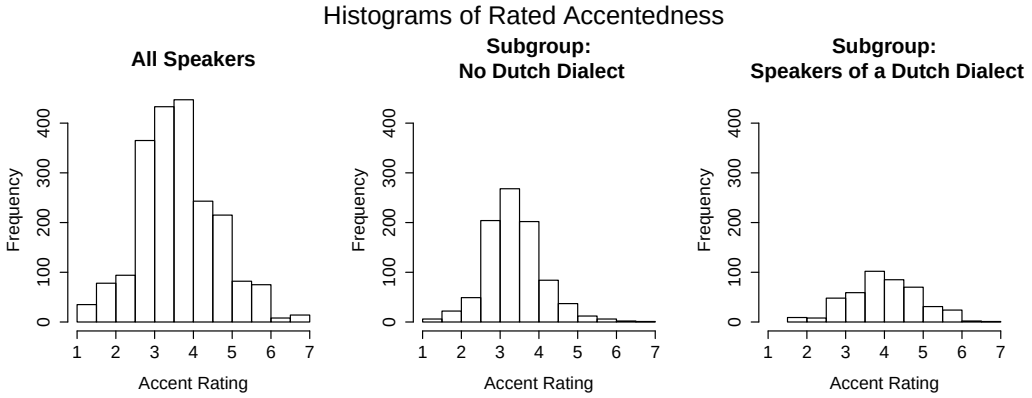


Figure 4: Distribution of participants' mean accentedness rating. Values correspond to listeners' agreement when asked whether a speaker had a strong accent. Subgroups rely on participants stating that they do or do not speak a Dutch dialect.

to express that they could not detect an accent in a recording. It could also have been a simple way of just skipping the task. In order to make sure that the speakers' average ratings are not just the result of random usage of the slider, a one-way between subjects ANOVA was performed in R (R Core Team, 2015) . The ratings significantly differed between speakers ($F(2088,22482) = 3.56$, $p < .001$).

2.3.2 Location Judgements

Among the tasks which required participants to make judgements about other participants, they were also asked to guess where another participant was from by pointing at the corresponding location on a map. They were prompted to do so by one out of three questions. It was asked where they thought the speaker was from, where they thought the speaker lived or whether they could place the speaker on the map[4]. Selected locations were stored as coordinates. Of all participants, 897 had their location guessed by a fellow participant. This number includes only judgements on recordings of

[4]The original questions in Dutch: "Waar komt de spreker vandaan?", "Waar denk je dat de spreker woont?", "Kun je hem/haar plaatsen op de kaart?"

spontaneous speech.[5] Furthermore, judgements referring to locations outside of the Netherlands were excluded. On many occasions, the same recording was guessed twice by the same participant. In these cases, these were collapsed to refer to just one location defined by their mean coordinates. With many participants having their location guessed more often than once, there were 2001 such location judgements made by 1064 participants.

2.4 Challenges for Accent Location

For accent location, the data prove to be challenging on several levels. A common problem with research projects especially with those using large scale data from anonymous sources concerns the reliability of the data. Since the project's application was freely available on the Internet, there was no control over how and by whom it was used. It cannot be guaranteed that all participants filled in the questionnaire correctly. On occasion, they may have misinterpreted a question or may not have been been honest in their answers. It can neither be ruled out that data are in parts confounded as a result of several people using the same account. The opposite case in which one individual uses several accounts is also possible. Taking a leap of faith, most data are assumed to be correct. A model trained on these should still be sufficiently resistant to such noise.

The project *Sprekend Nederland* has repeatedly been featured on Dutch television and has also been covered by other media. Its purpose including some research questions have been communicated to the public. As an experiment, the project's application corresponds to an open trial for which it has not just not been attempted to mask its background but participants have been explicitly informed. Much of the project concerns people's attitudes towards other people and their accents. The same attitudes may have influenced the participants' speech when they recorded themselves.

The dataset is not balanced with respect to location, age, gender and social background of the participants. With the exception of population distribution, the data are neither representative of the Dutch demographics. From a technical perspective, it is desirable to have a balanced dataset. However, means to counterbalance the dataset are extremely limited given the incompleteness of the meta data. In 4.3.4, ways of dealing with unequal

[5]For all judgements, the log-file registered the speaker's prompt. However, it was not possible to establish to which exact recordings judgements referred. The indicated number of location judgements is restricted to those based on recordings of spontaneous speech. This selection fairly ensures that judgements refer only to the longest recordings and allow for comparison with the models below, which were trained and tested on recordings of at least 10 seconds.

prior probabilities in training a model are discussed.

People with a higher education and professions of high status are often linked to the usage of standard Dutch, which is often thought to be a non-regional variety (Smakman, 2006). If the remaining participants for whom there is no information about their social background also happen to be mostly higher educated and have similar professions as those mentioned above, then the majority of speech samples can consequently be expected to contain little regionally accented speech. According to the participants' self-assessments at least, 41% should not show a regional accent whereas when judged by other participants they still may show some accentedness. All in all, there is reason to fear that a substantial part of the recordings does not contain speech with a regional accent.

3 Data Preparation

This section deals with preparing the data for training and testing models. Included steps cover data selection, feature extraction, partitioning and storing. Furthermore, results of a principal component analysis of the extracted features are presented.

3.1 Data Selection

It has been mentioned above that recordings of a duration shorter than 10 seconds were excluded from the beginning. The reason for this is that very short recordings are considered to insufficiently cover a speaker's variability. That is, short recordings are unlikely to contain all of the Dutch phonemes and are even less likely to contain all according allophones. It may be objected that the chosen minimum duration may still be too short to guarantee this. However, this value was chosen as a compromise between having too many recordings which badly represent a person's speech variability and having a drastically reduced dataset. On average, selected recordings ended up having a duration of 12.82 seconds.

3.2 Feature Extraction

Since it is not feasible to directly train models on audio data, features had to be extracted from the recordings which were meaningful for the accent location task. A popular choice in applications for speech and speaker recognition are MFCC features. Initially, i-vectors were used which were generated using the extractor from the Voice Biometry Standardization Initiative (Glembek,

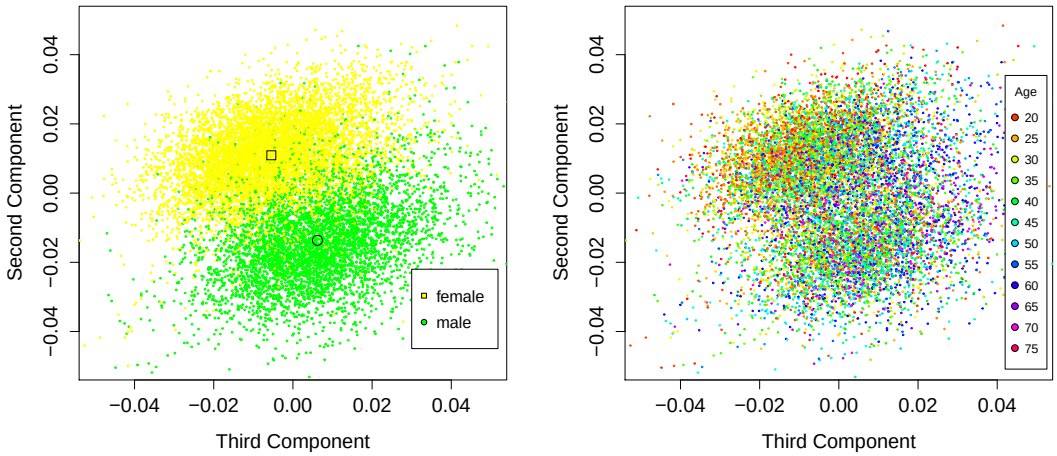
Burget, & Matejka, 2015). Their system uses a Universal Background Model (UBM) trained on telephone data and MFCC features extracted from the respective recordings. By using i-vectors, a recording formerly represented by a sequence of several MFCC features can be compactly represented by just one information rich vector. However, the fact that this system is optimised for telephone data required that recordings had to be transformed accordingly. This involved down-sampling to 8kHz and implied a loss of information. This motivated the creation of a customised i-vector system which was built by David van Leeuwen. Main differences are that audio data were re-sampled to 16kHz and bottleneck features were used in combination with MFCC features. The UBM had been trained on recordings of Dutch speakers from the *LUCEA* corpus (Orr et al., 2011). Per recording one i-vector was extracted resulting in a total of 13781 vectors. However, in training and evaluation of all models presented below, mean vectors were used which were created for speakers who provided more than one recording.

3.3 Principal Component Analysis

In order to estimate whether the i-vectors convey any information useful for the accent location task, a principal component analysis (PCA) was performed for which the Scikit-learn toolkit (Pedregosa et al., 2011) was used. Its main findings regard extra-linguistic information namely gender and age. The second component was found to strongly correlate with the gender of the participants (see Figure 5a).[6] However, such a pattern was expected to be found. The fundamental frequency is a very salient acoustic indicator for whether a speaker is female or male. With all the voiced intervals of the recordings there was abundant information about each participant's gender. The same two components were also found to be those which showed the strongest correlation with age. Although age related patterns are visible in Figure 5b it also shows that the relation between the component's eigenvalues and age is much less coherent.

Fewer cues about the participants' local accents were expected to be in the data. In fact, these were found to be extremely scarce. For each of the first 50 components, two linear mixed effects models (Bates, 2010; Kuznetsova, Brockhoff, & Christensen, 2015) were fitted using the eigenvector as dependent variable, the participant's identifier as random effect and either latitude or longitude as fixed effects. For a number of components, latitude or longitude reached statistical significance. In other words, a speaker's position did partially explain a component's values. However, their estimates were

[6]This only refers to 61% of the participants.(see Section 2.2.3)



(a) Eigenvalues coloured by gender. *(b)* Eigenvalues coloured by age.

Figure 5: Eigenvalues of the second and third components. Each point represents one recording.

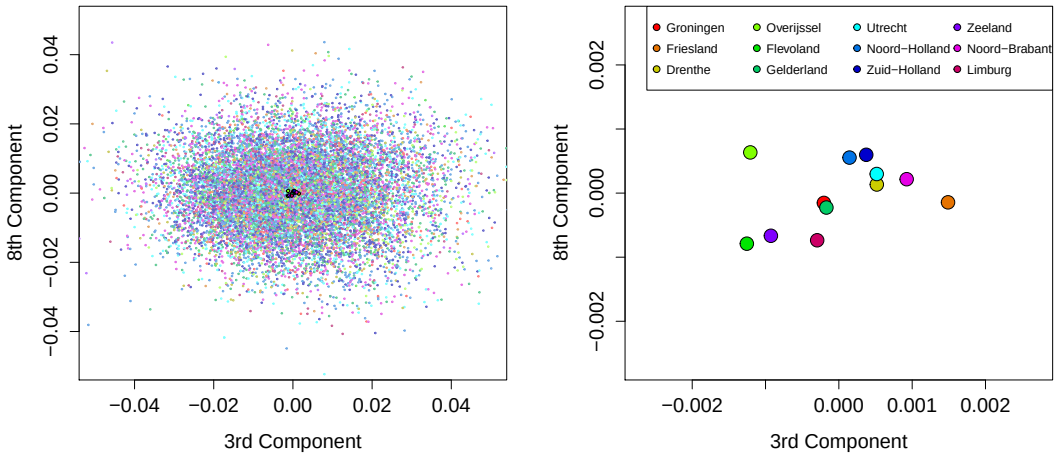
minute.

The highest estimate for latitude was found in the third and for longitude in the eighth component. The corresponding eigenvalues are shown in Figure 6a with points being coloured according to the corresponding province. On visual inspection, eigenvalues appear to be randomly distributed. Only if one zooms in to their mean values, small differences become apparent as Figure 6 shows.

To conclude the analysis, the i-vectors were found to convey information about individual speakers. However, cues about local accents were found to be scarce.

3.4 Partitioning for Cross-Validation

When models are build it is generally desirable to have separate datasets for training and testing. This can be considered a standard prerequisite for being able to find out how well a model generalises, i.e., how well a model performs on previously unseen data. A model's performance may be perfect if only tested on data on which it has been trained, however, it may perform poorly on new data. The question then is which data or rather which portion of the available dataset one wants to 'sacrifice' for testing the model. Especially if the available dataset is rather limited, one wants to



(a) Eigenvalues per recording. *(b)* Mean eigenvalues per province.

Figure 6: Eigenvalues of the third and eighth component. Points are coloured indicating the corresponding province. Dark bubbles in the centre of (a) correspond to mean values as depicted in (b). Means were calculated based on mean values per speaker.

reserve as much data as possible for training. However, a testing set which is too small may lead to less informative test results. The dataset available for this project as a whole may seem big, however, is in fact very limited, for instance, with respect to certain regional accents and age groups as has been discussed above. Therefore, the decision was made to prepare the data for cross-validation. That is training and testing were planned to happen repeatedly, and each time a different portion of the dataset would serve as testing data. That way it is possible to use just a small amount of data for testing. If the entire dataset is well balanced, models should be very similar. How well a model trained on the entire data would perform can then be inferred from the average test results.

The data were prepared for cross-validation by creating a vector carrying partition labels. All data points were labelled belonging to one out of ten partitions. Each partition was supposed to be about equally representative of the entire dataset. In order to achieve that, partitions were created such that they would all have a similar number of data points as well as similar proportions with respect to the regions they refer to. In other words, data points of one region were equally spread among all partitions. This way the dataset was balanced and in the cross-validation differences between models

and their test results could be minimised. However, partitions were not necessarily balanced with respect to other factors such as gender and age due to the above mentioned incompleteness of the meta data. Assigning data points of one region to a certain partition was done in random fashion. Importantly, however, if several recordings were available for one participant, they would all be labelled to belong to the same partition. This way the train and test datasets would never contain data from exactly the same source, i.e., an individual's recordings would never be used for both testing and training.

3.5 Storing

All data and labels were stored as arrays in separate files using Numpy (Walt, Colbert, & Varoquaux, 2011). All i-vectors were stacked vertically. Next to class labels, separate files were also created for the participants' identifiers, partitions, gender, accentedness ratings and coordinates. All were in parallel order to the data. These arrays are comparable to columns of one table. However, using separate arrays allowed for using the correct data type from the beginning (i.e., floating point numbers for numerical data and strings for labels).

4 Accent Location as a Classification Problem

This section is about building an accent recognition system which is able to predict a person's location outputting the region for which her or his speech is most typical. Treating accent location as a classification problem requires defining a set of regions. The assumption is made that people of each region have something in common in the way they speak which sets them apart from people of other regions.

The system was implemented as an artificial neural network. These types of models are very flexible and currently among the most popular tools for all types of recognition tasks. However, whether a network is able to make meaningful predictions largely depends on how it has been trained. Extensive experimentation was necessary to improve its performance.

In this section, the network's architecture is introduced. A description of the implementation also provides an overview of how repeated experimentation was made possible. The optimisation of the training process is discussed in more detail whereby effects of adjustments are illustrated using the output of several alphabetically named models. These were initially built to classify

speakers by their province. Having established ways to optimally train an accent classification network, the discussion will move to finding more suitable classes. The final results are then reported for these newly defined accent regions.

4.1 A Model for Classification

4.1.1 Neural Networks

The classification models discussed in this paper were all built as fully interconnected feed-forward neural networks using the TensorFlow package (Abadi et al., 2016) for Python. TensorFlow is an open-source software library for Machine Intelligence.

The network's architecture is described by an input layer of the size of the i-vectors (400), three hidden layers of 1000 sigmoid neurons each and an output layer of a size equal to the number of classes. That is, each output node corresponds to one class. The dimensions of the models were chosen such that they have a potential beyond perfectly fitting the training data. Some deeper alternatives were tried. However, finding the ideal architecture for the task was not deemed feasible within the project as the alternatives are countless in theory and in practice only limited by the specifications of the computer used for the task.

4.1.2 Alternatives to Neural Networks

Modelling local accents with neural networks was preceded by experiments with a number of different machine learning algorithms from the Scikit-learn toolkit (Pedregosa et al., 2011). These included Linear Discriminant Analysis, Naive Bayes Classifiers and Support Vector Machines. All of these worked to the extent that they were all able to make predictions which were consistently three to five percent better than random recall. These results were, of course, all far off a success rate of even a third of correct predictions. At that stage it was not yet clear whether this was due to the training data and selected features or whether these techniques were unsuitable for the task. Neural networks appeared to offer the most flexibility especially when dealing with unbalanced and sparse data. However, the results presented below are just somewhat better than what had been achieved with the above mentioned algorithms. Therefore, these may still be suitable alternatives. Training models with the Scikit-learn toolkit is also easier to implement in Python. It is possible to write scripts which are executed line by line whereas TensorFlow requires a graph to be built first.

4.2 Implementation

Repeated training and testing with different settings required flexible scripts which could be called from the Linux Terminal and thereby avoid necessary manual changes between trials. A main script for training and testing was created making use of a dataset class designed for managing how and which data are used for cross-validation.

4.2.1 Main Script

The main script was responsible for executing all necessary steps: data selection, preprocessing, model training and evaluation. It was designed to enable experimenting with a number of settings, which could be changed by using the respective arguments when called from a Terminal. Great care was taken to ensure data for training and testing would be kept separate and results would be replicable. Randomness was controlled for by reusing the same seed in all training sessions.[7]

Settings concerned data subselections, number of training iterations, initial learning rate, learning rate decay, size of mini-batches, dropout rate, balancing data, collapsing data, prior scaling and posterior scaling. The effects of applying changes to these settings will be discussed in more detail in 4.3.

The main script allowed for even more settings to be changed flexibly. For instance, the network's hidden layers could be changed. However, for the sake of comparability, the number of layers and their neurons were kept constant for this work. Furthermore, optional dimension reduction via a principal component analysis from the Scikit-learn toolkit (Pedregosa et al., 2011) was included as well. However, since models turned out to reach their best performance when trained on the original i-vectors, dimension reduction will not be discussed any further. All remaining settings were either also kept constant or ceased to be relevant for this work.

By rerunning the script, previously created models could be loaded, for instance, in order to evaluate their performance on a subset of the data.

The main workings of the script shall be presented in the following subsections.

4.2.1.1 Initializing Dataset

At first, a dataset object was created taking selected data, labels and partition labels as input. In case only a subselection of data was supposed to

[7]This way the same pseudo-random numbers were generated. Given that data and settings had not been changed, results from previous trainings could be replicated.

be used in training, the dataset was reduced accordingly. Mean vectors for speakers who had provided several recordings could be generated by collapsing the data accordingly.

4.2.1.2 Preprocessing

With each round of the cross-validation, the selected training data were preprocessed. They were normalised to zero mean and unit variance. Normalisation on the test data was performed using the mean and standard deviation of the training data. Means and standard deviations from each cross-validation round were saved for possible retesting of the models.

4.2.1.3 Training and Evaluation

The cross-validation was partly parallelised. Training, evaluation and saving the models happened in separate processes. Each time a new graph was created and each process made use of a deep copy of the dataset object. However, each time a different partition was selected to serve as test data.

Settings regarding training were partially hardwired. Training was prepared by setting weights to random numbers sampled from a normal distribution with a standard deviation of 0.05. The optimisation algorithm was set to Gradient Descent and the cost was defined as the mean cross entropy after applying the softmax function to the output.

Training was implemented using two nested loops. The outer loop iterated over epochs. With each epoch the learning rate was reduced depending on the selected value for learning rate decay. With each step of the inner loop the cost for the given training data was calculated and the network's weights were updated accordingly. If all training data were used per update (i.e., in full batch learning), the inner loop consisted of just one step. More steps were required if the training data were fed in smaller batches. Their number then depended on the selected batch size.

For testing the network, data reserved for this task were fed into the network. The class corresponding to the output node with the highest activation was counted as prediction.

4.2.1.4 Saving Output

After training had finished, the remaining steps consisted of calculating a mean confusion matrix from the confusion matrices of each cross-validation round and finally saving all matrices.

4.2.2 Dataset Class

For classification, a Dataset class was written with which data and labels were handled. As input, it used before mentioned Numpy arrays. On initialisation, parallel representations were created for labels as numeric arrays as well as corresponding arrays of one-hot vectors. For cross-validation, all data and labels belonging to one partition were stored as copies for testing and all remaining data and labels were stored as copies for training. This was done iteratively such that each partition was used once as testing set.

A selection of added features are introduced here. Some of these were used in optimising the models and will be discussed in more detail below.

4.2.2.1 Subselections

The dataset could be reduced to a subselection if one or more boolean arrays were provided indicating eligible data points.

4.2.2.2 Class Prior Probabilities

For each cross-validation round, prior probabilities were calculated based on the frequency of class labels in the training data. An array was created indicating the corresponding prior probability for each data point.

4.2.2.3 Mini-Batches

Training data could be returned in so-called mini-batches. Batches of the desired size could be returned including corresponding labels and prior probabilities. When using mini-batches, the class kept track of the number of training iterations, i.e., it counted how many times the complete training data had been returned. On each iteration the training data were reshuffled.

4.2.2.4 Balancing Training Data

It was possible to balance the training data such that there were the same number of data points for each class. This was achieved by sampling training data from each class based on the size of the smallest class. Sampling could optionally be repeated on each training iteration such that eventually all data would be used.[8] Given the requested batch size was a multiple of the number of classes, it was then also possible to have fully balanced batches returned. That is, in each batch there were the same number of data points per class.

[8] Alternatively, smaller classes could be expanded to the size of the biggest class by filling these with copies.

4.2.2.5 Collapsing Data

Given a vector of additional labels, data points of the same label could be collapsed to their average. For instance, one mean vector could be created per speaker if she or he had provided several recordings. This option was used in training all models which are presented below.

4.2.3 Hardware

The main scripts for this work (e.g., feature extraction and model training) were computationally very demanding and could not have been run efficiently on an average laptop. Therefore, it was necessary to run these on an external computer cluster with sufficient processing power and random access memory. By courtesy of the Centre for Language Studies of Radboud University Nijmegen, access was granted to the Equestria cluster [9].

4.3 Optimisation

Whether a neural network gives any useful output highly depends on how it has been trained. Especially with this project for which data were unbalanced and often sparse for certain regions, finding the right parameter settings was crucial. Several methods which potentially help to train the networks such that they generalise well were repeatedly tested with different settings. In 4.2, these methods were introduced. In this section, it will be shown how training was affected by applying these methods. The network's output will be evaluated on recall and precision. The former informs about the portion of recordings belonging to one class which were correctly classified. Precision refers to the portion of recordings predicted to belong to the same class which actually are from that class.

If one wants to illustrate how different ways of training affect a network's performance, its architecture should, of course, not be changed between trials. Generally, the most meaningful comparisons can often be made if one keeps all but one parameter constant between trainings. More than 400 such trials were conducted obeying this principle. However, it is virtually impossible to train as many networks as there are different possible settings.

In this section, a small selection of models of before mentioned trials serve to illustrate how different training methods affected performance. When showing examples of networks the singular is used for simplicity. However, this in fact refers to ten networks created during a cross-validation. Results

[9] also referred to as Ponyland

always refer to average values of the output of all ten networks. For illustrative purposes, exceptions are made when discussing the learning progress. In that case the first model created during a cross-validation is used.

4.3.1 Learning Rate and Learning Rate Decay

The learning rate defines how much the network's weights are changed with each learning step. Fortunately, there usually are several learning rates which will all enable a network to learn. Higher values result in faster learning. However, if the learning rate is chosen too big, it will result in an increasing error (Bengio, 2012). On the other hand, no learning may be observable within the chosen number of epochs if the learning rate is too small.

When training a network on batches of data it is always necessary to reduce the learning rate over time (Wilson & Martinez, 2003). Reducing the learning rate has the consequence that weight updates become more and more subtle. That way it can be achieved that a weight's value approaches the ideal value (true gradient) for the given data. Keeping the learning rate constant would have the effect that weights oscillate around ideal values with each update leading to either overshooting or undershooting. For this work, the learning rate was reduced exponentially. After each epoch of training, the learning rate was multiplied by a chosen factor < 1.

Figure 7 illustrates how different settings of initial learning rate and its decay affected the final performance of a network trained with full batches throughout 1000 epochs. In other words, one learning step was applied per epoch using the entire training data. The figure shows that if the learning rate is reduced to fast, the network does not learn enough to make correct predictions. Reducing the learning rate by factors 0.95 and 0.995 led to near random performance whereas the slowest decay resulted in somewhat better performance. However, reducing the learning rate by factor 0.9995 over 1000 epochs meant that the last training step used a learning rate which was still 60% of its initial value, which is potentially still too much. It is therefore worth having a closer look at the best performing network, which will be referred to as Model A. In order to track the networks performance throughout training, the first model created during a cross-validation was tested on the training and test data after each completed epoch. Figure 8 shows how much of the data were correctly classified. [10] It shows that updates still had a big effect towards the end of training. Moreover, it appears

[10]Note: All data were used for monitoring the training process. Since classes are unequally represented in the dataset the recognition rate always reflects especially the network's performance on bigger classes, i.e., it is not identical to the mean recall reported elsewhere in the text.

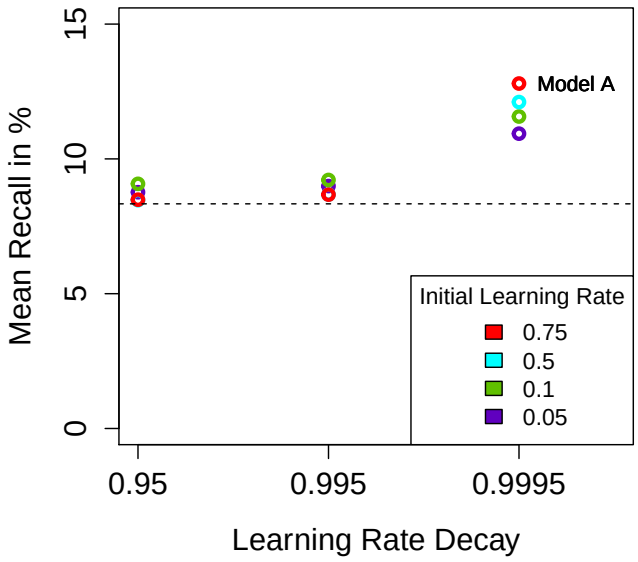


Figure 7: Mean recall of classification networks trained with full batches. The dotted line indicates random performance, which is equivalent to correctly classifying one out of twelve samples.

that a prolonged training would create a better fit. This, however, was not favoured as one cross-validation of a network trained with full batches already took about six hours to complete.

A quick look at Model A's precision and recall per class (Figure 9) reveals yet another problem. It is not just that recognition is overall poor. In fact, several classes are entirely ignored. The network especially zooms in to classes for which there are more recordings available (compare with Figure 2 above).

Due to these performance issues, however, especially due to the fact that training was extremely slow, training with full batches was abandoned.

4.3.2 Training with Mini-Batches

Faster training was possible using so-called mini-batches which also lead to more meaningful results. This way of training differs from the above in that instead of using the entire training data per learning step, only a random sample is used. Sampling happens without replacement. One epoch consists of several learning steps and is completed as soon as the entire training data

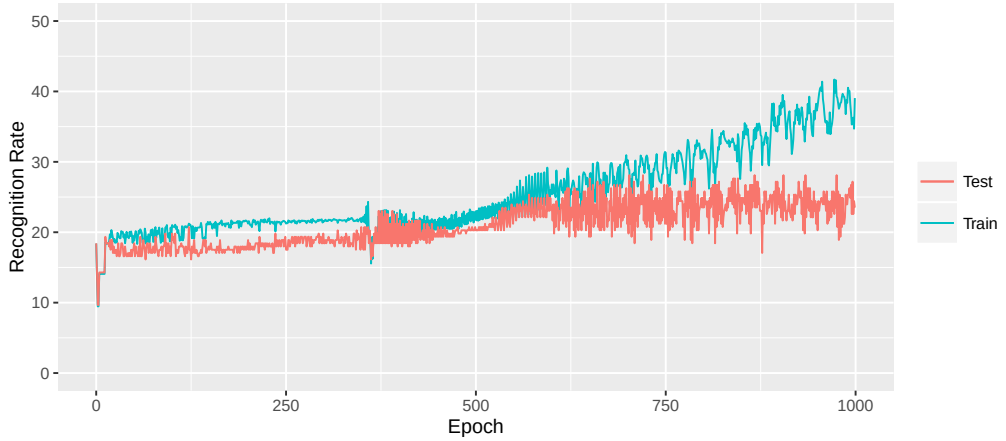


Figure 8: Accuracy of Model A during training with full batches. The recognition rate indicates correct classifications of all training and all test data. This refers to the first out of ten models trained during a cross-validation.

have been used once.[11] In training Model B, batches consisting of 36 i-vectors were used. The initial learning rate was set to 0.5 and reduced by a factor of 0.99. This faster reduction of the learning rate was found to work since training progressed faster in comparison to training with full batches, i.e., more learning steps were made per epoch.

On average, Model B was able to correctly classify 12.68% of the test data with a precision of 12.90%. The mean recall was similar to that of Model A. However, Model B did not ignore entire classes. A previously observed problem remained. Bigger classes were still dominant. Section 4.3.4 will show different techniques which were applied to deal with this issue.

A special case of training with mini-batches is on-line training in which batches of just one data point are used. Consequently, one epoch consists of as many learning steps as there are data points. Bengio (2012) lists a number of advantages of this approach over batch learning. The strongest argument is that faster convergence is achieved with on-line learning.

Indeed, this way of training allowed for reducing the number of epochs to less than a third of those 1000 epochs used otherwise and even slightly improved accuracy. In practice, however, the overall increased number of learning steps resulted in much longer training. Nevertheless, it was decided to train all subsequent models in this manner (i.e., Models C - G).

The observed increase in training duration, however, should not be inter-

[11] In practice, an epoch was counted as completed if not enough training data were left making up the selected batch size.

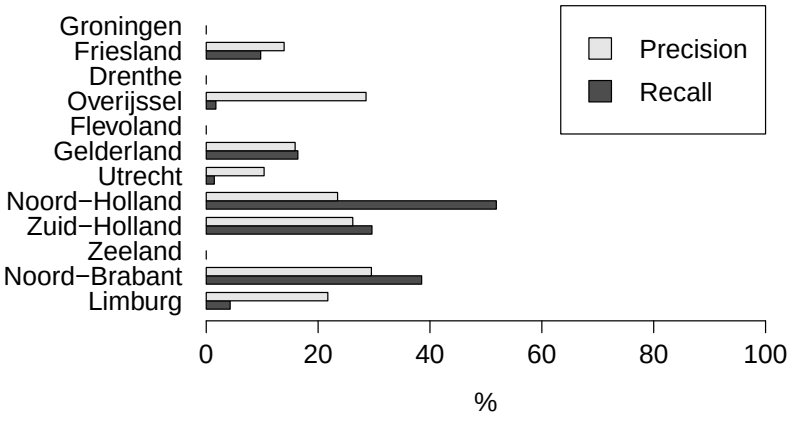


Figure 9: Precision and recall of Model A.

preted to disprove Bengio (2012). It can merely be attributed to how training was implemented.[12]

4.3.3 Dropout

Figure 11 shows Model B's ability to classify training and test data after each of the first 200 epochs. It reveals a major problem which is often encountered when working with neural networks which is over-fitting. At the beginning of training, the network improved on both training and test data. However, while the network continued to improve on the former as far as making perfect predictions, the network showed no improvement when test data were used. Quite opposite, it even became somewhat worse. In other words, the network badly generalised.

According to Hinton, Srivastava, Krizhevsky, Sutskever, and Salakhutdinov (2012), over-fitting happens if training data are sparse and a network has enough hidden neurons which allow to perfectly model training data with many different settings. Their proposed technique for reducing over-fitting, which is commonly referred to as dropout, prevents complex co-adaptations on the training data. This works by randomly omitting a selected number

[12]The actual training happened within the TensorFlow graph, which could be expected to proceed as efficient and fast as it is currently state of the art. This includes estimating the gradient from batches of data, which becomes computationally more demanding with larger batches. However, data were fed into the graph from outside. It is presumably these feeding steps (including retrieving data from the dataset) which delayed computation.

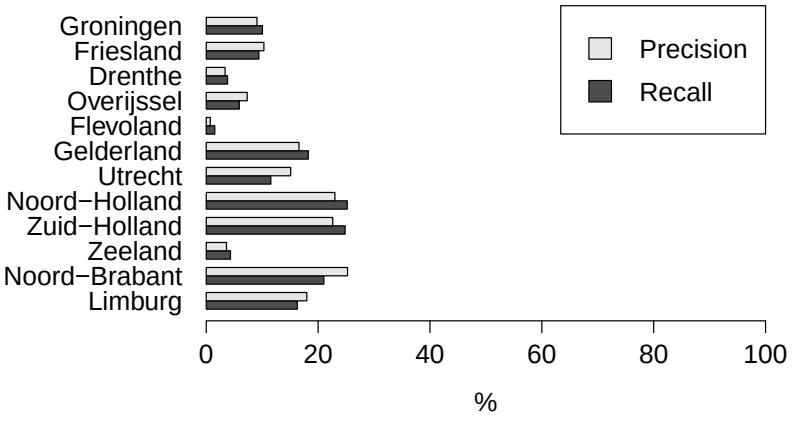


Figure 10: Precision and recall of Model B.

of hidden neurons during training. With each new learning step, previously omitted neurons are recovered and a different random selection is made for omission. This is often compared to training a large number of networks and using their average results in prediction. However, dropout requires much less computation.

In practice, dropout was applied to the last hidden layer. Figure 12 shows the accuracy of Model B* during training again. It was trained as before except for making use of dropout and training with the same number of i-vectors per class (see 4.3.4)[13]. The dropout rate was set to 0.25, which is equivalent to the probability of a neuron to be kept during training. In contrast to Model B (see Figure 11), the network kept improving on both training and test data.

4.3.4 Prior Class Probabilities

The output of above mentioned networks Model A and B indicates that both had more difficulties with correctly classifying data from smaller classes. In theory, this could be due to these data being less discernible. This would imply that the correlation of class size and the network's abilities was mere coincidence. An alternative explanation, however, would be that the networks had more difficulties with learning from smaller classes. The former explanation would imply that nothing can be done to alleviate the problem

[13]Applying dropout to Model B without accounting for different prior class probabilities just delayed over-fitting.

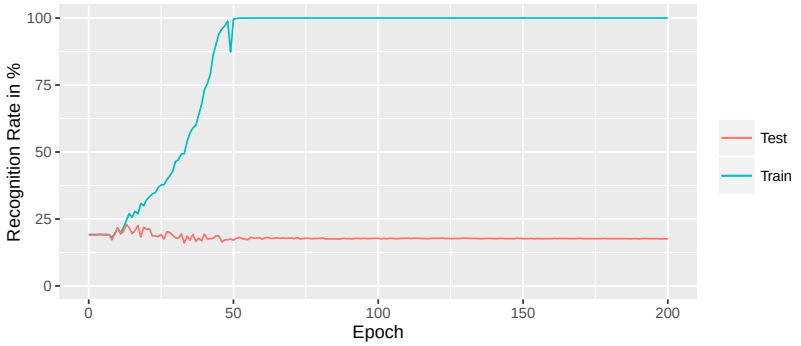


Figure 11: Accuracy of Model B. Of a total of 1000 epochs, the first 200 are shown. The recognition rate indicates correct classifications of all training and all test data. This refers to the first out of ten models trained during a cross-validation.

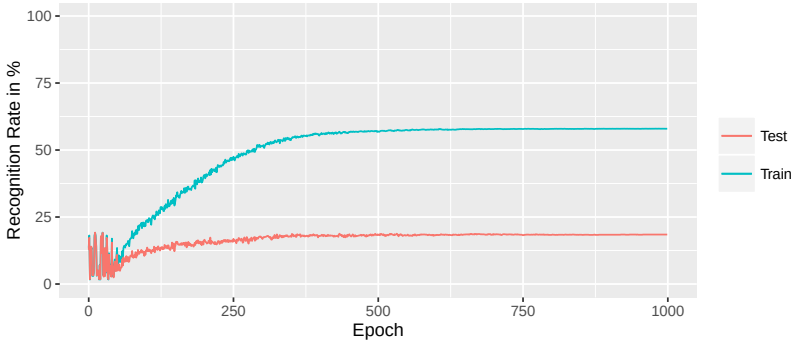


Figure 12: Accuracy of Model B* with applied dropout. The recognition rate indicates correct classifications of all training and all test data. This refers to the first out of ten models trained during a cross-validation.

within the limits of the available dataset. The latter explanation, however, corresponds to a phenomenon which has repeatedly been observed and for which there are solutions. In Lawrence, Burns, Back, Tsoi, and Giles (1998) a number of such measures are discussed. The following subsections will shortly introduce some of these and their application.

Apart from purely improving the model, there is another good reason for aiming at avoiding varying class prior probabilities to influence training. When a model for accent location is evaluated one is usually interested in knowing how well a model is at actually detecting an accent purely based on the input. If, however, predictions about class membership of some data are conflated with the classes prior probabilities, one may draw incorrect

conclusions about a models detection ability. That is, one may confuse the latter with the model's acquired knowledge about demographics.

4.3.4.1 Subsampling

In order to achieve that all classes are represented evenly in the training data, one may choose to reduce data of each class to the size of the smallest one. With the exception of the latter, each class is represented by a smaller sample. If this is just done once before training, this method implies data loss.

Similarly, classes of equal size can be created by expanding all classes to the size of the biggest class. This can be achieved by adding sampled duplicates. Its advantage is that no data is lost. In trials, this method led to similar results as repeated subsampling (see 4.3.4.3 below)

4.3.4.2 Probabilistic Sampling

Probabilistic sampling is another method which works by changing the input of training data. It is defined by Lawrence et al. (1998) as a loop randomly selecting one class followed by random selection of one data point.

4.3.4.3 Repeated Subsampling

What is called repeated subsampling here is an own approach which can be considered as a combination of the previous two methods. It involves repeating subsampling after each completion of an epoch (see implementation in 4.2.2). A reduced dataset is used per epoch. However, over several epochs this does not result in data loss.

Model C was trained in that manner, which had the desired effect as Figure 13 shows. Furthermore, this model was trained throughout 300 epochs in on-line manner (mini-batches of size 1). The initial learning rate was set to 0.0075, learning rate decay to 0.99 and dropout to 0.90. With the exception of learning rate decay these settings were also applied to all following classification models.

In combination with before mentioned dropout, mean recall and precision reach 17.83% and 15.97% respectively.

4.3.4.4 Post Scaling

When using post scaling networks can be trained as usual. Instead of manipulating the training data it requires the network's output to be corrected. In practice, the output neurons' activations are divided by the corresponding

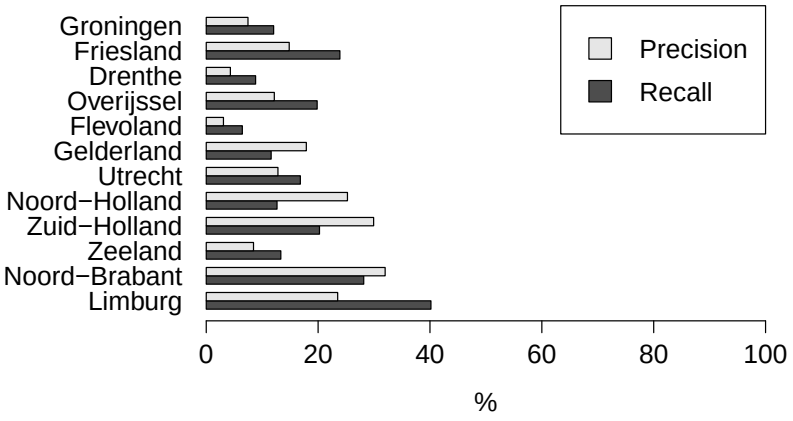


Figure 13: Precision and recall of Model C.

class probabilities. This results in activations of neurons corresponding to smaller classes to increase. In practice, this only works if activations do not differ too extremely. In other words, the network needs to have learned about all classes at least to a certain degree. For instance, Model A is such an extreme case where neurons of the smallest classes were barely activated such that they would keep being ignored if one applied post scaling (see Figure 9).

4.3.4.5 Prior Scaling

Prior scaling affects weight updates during training. This was implemented by increasing the cost for data points from small classes. In principle, one can just divide the cost by the according class prior probability. However, in this simple form where probabilities are numbers between 0 and 1, the overall cost becomes inflated as in fact all cost values end up as multiples of their previous values. Therefore, the prior probabilities were scaled to mean 1. Consequently, the cost for data from smaller and bigger classes became increased and decreased respectively.

This approach turned out to be successful with respect to smaller classes. However, the results were neither ideal since recognition rates for bigger classes dropped considerably. Lawrence et al. (1998) point out that prior scaling may be even more successful if its scaling factor is between 0 and 1, i.e., no prior scaling and full scaling. Scaling with factors other than these two extremes was implemented by multiplying the fully scaled cost by

the scaling factor and adding the non-scaled cost multiplied by 1 minus the scaling factor. Model D made use of prior scaling. The approximately ideal scaling factor was found to be 0.95. In contrast to Model C, all training data were used per epoch. Therefore, the learning rate decay was set to 0.95.

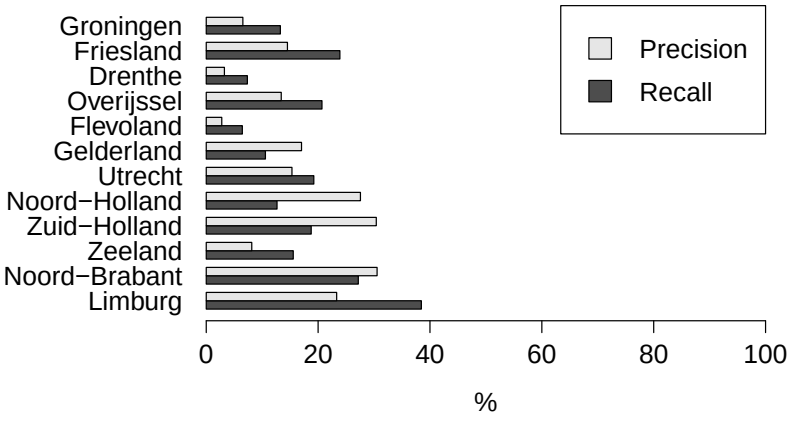


Figure 14: Precision and recall of Model D.

Model D's mean values for recall and precision are 17.84% and 16.07% respectively. Overall, the performance of this model and Model C are almost identical.

4.3.5 Training with Selections of Data

4.3.5.1 Gender-Dependent Models

In speech recognition, systems have been shown to perform better when gender-dependent models are used (e.g., Woodland, Odell, Valtchev, and Young (1994)). The principal component analysis in 3.3 has shown that gender accounts for a main part of the variation in the i-vectors. However, this variation does not help with the accent location task. The question is whether as in speech recognition it is a nuisance factor. In that case, using separate models can be expected to show better performance.

Before this approach could be tested, another problem had to be solved which is that 39% of the participants did not state their gender (see 2.2.3). In order to avoid data loss, labels indicating whether a recording belonged to a female or male speaker had to be generated. This was achieved by first

training a neural network [14] on the data of those participants for whom this information was available. It was able to correctly classify 97.6% of the data in a cross-validation.

The gender-dependent models were trained with the same settings as Model C. However, both performed worse. Recall and precision for the model trained on i-vectors labelled as corresponding to females were 15.97% and 14.67%. These values were even lower for the male model with 15.23% recall and 13.38% precision.

Training separate models has the disadvantage that only about half of the data can be used for each of the two models. If at all variation attributed to gender is actually problematic for accent location, the disadvantage of having to use two reduced datasets did not justify training separate models.

4.3.5.2 Discarding Unaccented Data

Data which show no characteristics of their own class are problematic in training models for classification. In practice, recordings of speakers of standard Dutch are of little use in modelling local accents. In fact, they lead to more confusable classes and subsequently to worse results.

Judgements on the degree of accentedness were available for almost all participants. As was discussed in more detail in 2.3.1, these were not necessarily reliable.

Several models were trained using training data corresponding to an accentedness rating above a selected threshold. Whether discarding supposedly unaccented data resulted in overall better performance depended on the method used accounting for different prior class probabilities. The biggest improvement was observed when Model C was retrained on data with an accentedness rating above 1.5. Recall improved by 0.19% and precision by 0.38% to 18.02% and 16.35%. In training Model C, repeated subsampling was applied. However, no sub-selection of training data could be found which would result in Model D, which made use of prior scaling, to show a better performance.

It could not be established whether the small improvement in Model C

[14]The model for gender classification was trained over 1000 epochs on mini-batches of 100 i-vectors applying repeated subsampling. Initial learning rate and learning rate decay were set to 0.7 and 0.99 respectively.

was actually due to using supposedly more meaningful training data.[15] If that was the case, it would need to be explained why the same approach failed with regard to Model D. Since there appeared to be little to gain from removing data based on accentedness ratings, this approach was deemed unsuccessful.

4.4 Alternative Classes

The first attempts at classifying accents as presented above made use of provinces as labels, i.e., models were built to return a participant's province given some speech. At first, the assumption that provinces have their corresponding accents may seem justified. For instance, this seems to hold true for the southern province Limburg, which has its own dialect called *Limburgs*. However, a valid objection is that borders of provinces have not been established based on how people speak but on politics. Figure 15 shows a mapping of Dutch dialects and provinces. The two rarely coincide. In fact, this is also true for the before mentioned dialect *Limburgs*, which corresponds to the red area covering both Dutch and Belgian territory.

Concluding that provinces are unlikely the best class labels it was attempted to find more suitable ones. It would suggest itself to simply rely on a mapping such as that by Daan and Blok (1969). However, with the available data, this would be unlikely to work well. Some accents would have to be modelled based on utterances of just too few people. This could only be alleviated by grouping the affected accents.

4.4.1 Clustering: Finding Divisions in the Population Distribution

Any division of the country into accent regions defined by strict boundaries will require some abstraction. Overall, accents do not change as abruptly as boundaries imply. Two speakers of neighbouring regions living close to

[15] A possible, however, untested explanation regards randomness. In training Model C, removing training data also influenced sub-sampling. In order to be able to replicate trials, randomness was controlled for by reusing the same seed. However, if some data were removed from a set, there was no guarantee that of the remaining data the same data points would end up being selected again. For instance, if the pseudo-random rule for sampling from some data points (e.g., A,B,C,D) includes always selecting the third item and a preceding item is removed in another trial, two different items will be selected in both trials (C and D).

In order to test whether the improvement of Model C was due to this issue, one would need to cross-validate all models with several different seeds and average over their outcome.

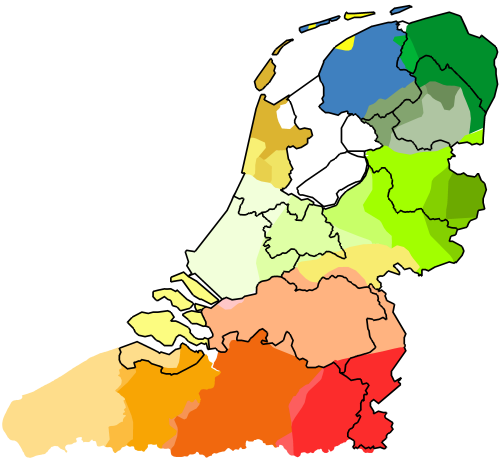


Figure 15: Dutch dialect map according to Daan and Blok (1969) (Map created using *Kaart package* (2013))

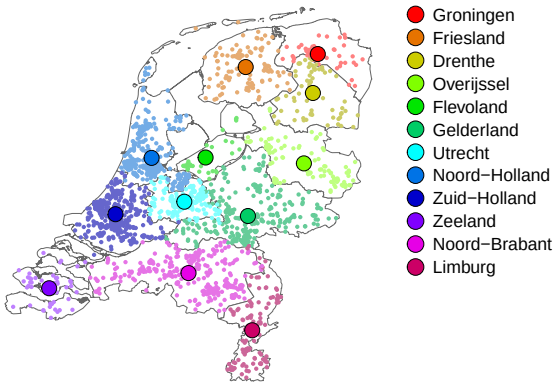
the shared border will often have similar accents. These similarities may be even stronger than those between these speakers and their regions' average accents. A recording of such a person will then be ambiguous for a classifier or human listener since it will appear equally likely to correspond to an accent of either of the two neighbouring regions.

Following this reasoning, new classes were created taking the population distribution (or rather participant distribution) into account. It was aimed at finding class boundaries along which the population density is low and thereby reduce the number of before mentioned ambiguous cases.

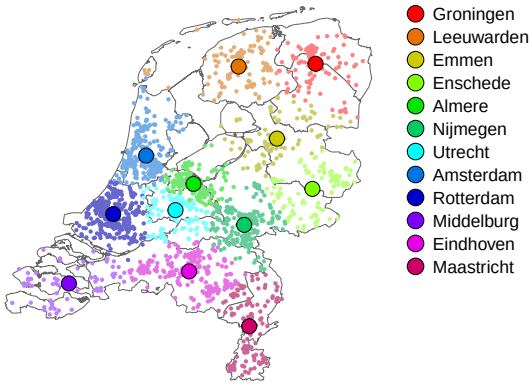
New classes were generated by taking the participants' locations and applying K-means clustering. Several locations were chosen as initial class centres (centroids). For each participant's location, the algorithm checked which would be the closest centroid. Locations of several participants were assigned to their closest centroids resulting in temporary classes. The centroids' locations were then recalculated such that they represented the mean locations of their assigned participants. After that, this process started again with each participant being newly assigned to its closest centroid and centroids being moved subsequently. These steps were repeated until no changes in assigning participants nor the location of the centroids occurred.

To illustrate how this approach differs from using provinces as classes, twelve new classes were created using the locations of the biggest cities of each province as initial centroids. The resulting clusters resembled agglomerations in a very broad sense.

As Figure 16 shows, the newly generated classes represent compacter



(a) Participant distribution by provinces



(b) Participant distribution by agglomerations

Figure 16: Locations of participants coloured according to class membership. Bigger dots mark the average locations (centroids) per class. Agglomerations are named after the centroids' locations previous to clustering. (These and all subsequent maps have been created using *tmap* (Tennekes, 2016).)

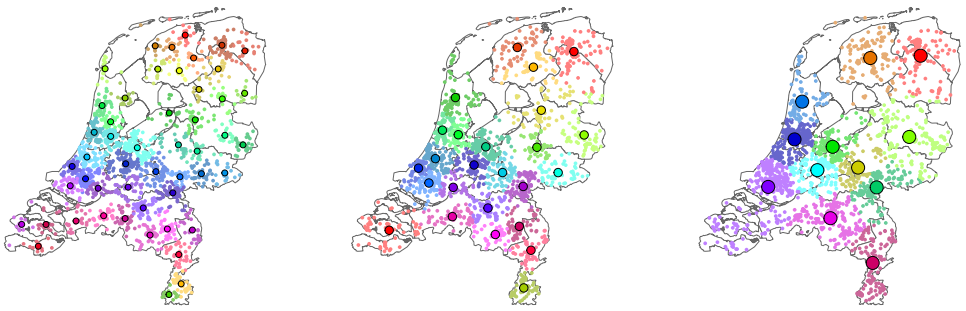
regions. Boundaries between two classes always form a straight line. (Compare, for instance, the borders of provinces Utrecht and Gelderland to those between agglomerations Utrecht and Nijmegen.) Furthermore, boundaries between two classes are drawn in places where the population density is comparably low. Although clusters are merely based on participants' locations they would likely correspond to clusters based on the actual population distribution. At least this would match with 2.2.1, where it was observed that the number of participants per province correlated with the provinces' populations.

The disadvantage of this approach is its blindness to geography. This becomes especially apparent with regard to the mapping of participants be-

longing to Leeuwarden. Furthermore, the way the clustering was initialised led to results which are not necessarily optimal since it still implies some dependence on provinces. Initializing the clustering with the provinces' biggest cities as centroids was merely done in order to be able to directly compare previously used provinces to newly generated classes. One may alternatively initialise the clustering with random centroids. This leads to clusters which are fully independent of any pre-existing division. However, this also leads to different clusters on each repetition which are neither always optimal.

4.4.2 Repeated Clustering

The solution found to the problem described above involved repeated clustering with a larger number of centroids and iteratively removing the centroids attracting the smallest number of points. Initial centroids corresponded to evenly spaced locations on the map whose number was a multiple of the desired classes. An outer loop was added to the algorithm described above. With each iteration, clustering proceeded as above. After convergence, the centroid attracting the fewest points was dropped and the remaining centroids were used in the next iteration. These steps were repeated until reaching the desired number of classes.



(a) 50 clusters *(b)* 25 clusters *(c)* 12 clusters

Figure 17: Clusters emerging from repeated K-means clustering whereby after each convergence the centroid of the smallest cluster is removed. Larger bubbles indicate the centroids' locations.

Figure 17 shows two intermediate states and the final state of the clustering. The result, represented by 17c, is a clustering which entirely relies on the participant distribution.

4.4.3 Manually Corrected Clusters

Taking the history of the country, the literature about its dialects as well as the intermediate results above into consideration, it was decided to dissolve one cluster and force the creation of a new one at a different location. The cluster in question largely corresponded to the region around Almere which is also rather close to Utrecht and Amsterdam (compare with Figure 16). Flevoland, of which Almere is the biggest city, is the newest province of the Netherlands. At the time Daan and Blok (1969) mapped Dutch dialects, only a small island in its north could be included. These days, the province is occasionally subject of research on emerging accents. However, accents spoken in Flevoland cannot be expected to be very distinct from its neighbouring accents. In fact, the intermediate results above show the worst precision and recall for this province. Due to its small population no cluster 'survived' which would correspond to Zeeland in the south-west of the country. A centroid corresponding to its biggest city was added to the remaining eleven centroids and a simple K-means clustering was performed again.

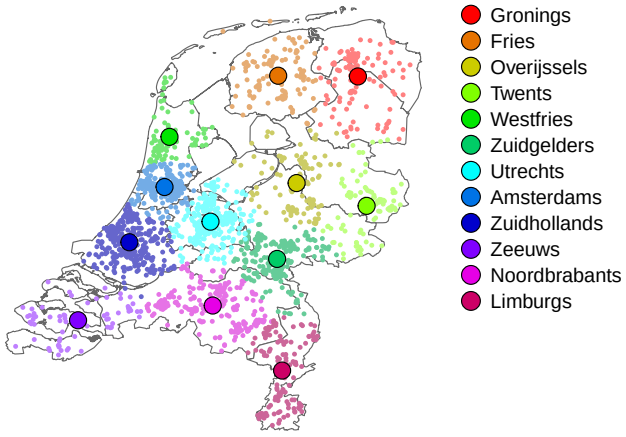


Figure 18: New classes resulting from repeated K-means clustering and some manual corrections.

The resulting clusters were subsequently compared to the dialect regions established by Daan and Blok (1969). Clusters were named after the dialect regions they best represented. In Daan and Blok, the region corresponding to *Zuidhollands* stretches out more to the north and also includes Amsterdam. However, separating the northern part and labelling it after the city dialect *Amsterdams* can be justified by the clustering results and the fact that this region is best represented in the dataset.

4.5 Results

The final classification results refer to the output of Model E which made use of the new classes defined above. It was trained just like its predecessor Model D.

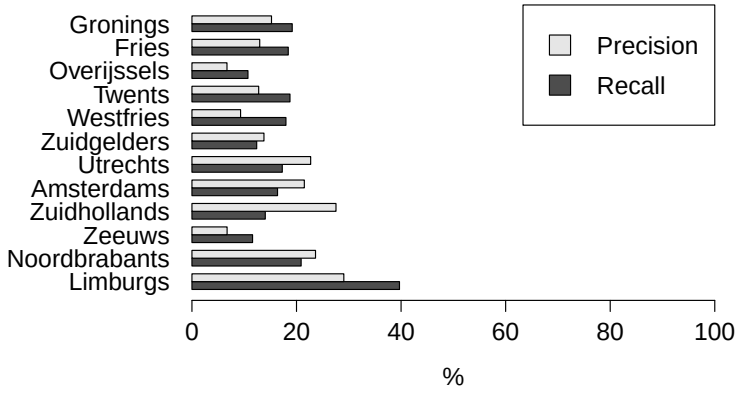


Figure 19: Precision and recall of Model E.

Overall, the model showed the best performance with an average recall of 18.11% and a precision of 16.83%. In contrast to all preceding models, all classes reached a recall exceeding the value corresponding to random performance, i.e., 8.34%. As figure 19 shows, there was also less variation in performance with regard to different classes. (Recall per class had a standard deviation of 7.54% whereas the second best model reached 9.07%.)

Still despite all efforts made for optimizing training and finding more suitable classes, the results clearly show that the vast majority of recordings could not be classified correctly. The confusion matrix (Table 1) shows that there were examples of all possible ways of misclassification. However, it also shows that recordings were more likely to be classified to correspond to a neighbouring region. A positive correlation was found between the number of false positives per accent and whether accents were neighbouring or not, $r(130) = .41$, $p < .001$. The following section will in part address the question how this observation can be exploited.

Table 1: *Confusion matrix for Model E. Rows indicate the accents to which the recordings actually correspond. Columns indicate the accent as which a recording was classified.*

	G	F	O	T	WF	ZG	U	A	ZH	Z	NB	L
Gronings (G)	**28**	11	13	12	5	15	11	14	10	9	13	5
Fries (F)	12	**21**	5	7	16	10	6	9	7	9	8	4
Overijssels (O)	8	9	**12**	9	9	8	13	13	7	7	7	10
Twents (T)	20	9	8	**18**	5	5	2	7	3	4	13	2
Westfries (WF)	8	15	6	2	**16**	3	7	9	11	3	6	3
Zuidgelders (ZG)	12	12	27	17	8	**27**	14	15	12	13	40	21
Utrechts (U)	20	16	23	15	32	21	**52**	37	35	12	20	18
Amsterdams (A)	27	22	16	15	22	26	36	**46**	38	11	13	9
Zuidhollands (ZH)	23	22	39	23	40	27	60	47	**54**	25	16	9
Zeeuws (Z)	7	6	8	2	5	7	7	3	5	**8**	6	5
Noordbrabants (NB)	16	13	19	14	10	32	17	12	11	12	**52**	41
Limburgs (L)	3	6	3	7	4	15	4	2	3	6	26	**52**

5 Accent Location as a Regression Problem

A potentially more precise approach to accent location will be introduced in this section. In changing the task to a regression problem, predictions refer to precise locations expressed by coordinates.

In the classification networks above, the output layer consisted of twelve neurons of which each neuron was associated with one class. In prediction, a recording was said to belong to the class for which the corresponding neuron showed the highest activation. Consequently, all remaining output was ignored even if another neuron showed similarly high activation. Given the apparently highly confusable data, there were many such border cases in which one class would win over another class by a marginal difference. In general, a classification does not account for gradual changes between classes. Therefore, these types of models are not good at representing the continuous aspects of accents. This problem can be solved if instead of regions precise locations are predicted by a model.

This section first deals with how the output of the classification model can be used to approximate a speaker's coordinates. Moving on to building a new regression model trained on i-vectors and coordinates, necessary changes to the network's architecture and training are introduced.

5.1 Reinterpretation of Classification Model's Output

A first attempt at locating accents in a continuum involved using all output of the classification model. By applying the softmax function to the output layer of the network, the converted output could be interpreted as a probability distribution. In other words, the output was changed such that it represented the probability of a recording to correspond to each class. Very low probabilities were regarded as little meaningful. In order to lower their impact, the probability distribution was smoothed accordingly. Smoothing was implemented by taking the square of the probability distribution and subsequent rescaling to its previous range (0-1).

Classes were reinterpreted as representing locations instead of regions. Each class got assigned the coordinates of its centre. A prediction was made by multiplying all centre coordinates by their according probabilities and taking the sum of these values. For instance, if the probability distribution showed a value of 0.5 for *Utrechts* as well as for *Amsterdams* and all other classes being zero, the recording was predicted to correspond to a location situated exactly between the two accent regions' centres.

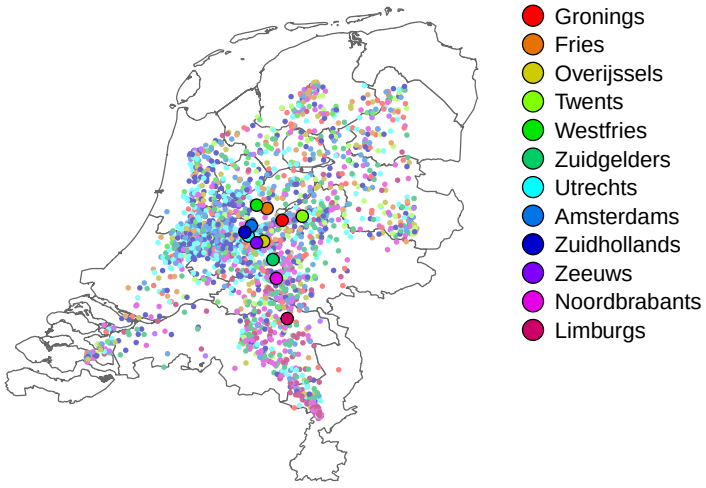


Figure 20: Predicted locations generated from output of Model E. Colouring corresponds to the actual accent region. Larger bubbles indicate the average predicted location per accent.

Figure 20 shows the predicted locations generated by applying this approach to Model E. Obviously, most recordings were predicted to correspond to locations between the classes' centres. Although the output reflected gradual differences between recordings of different accents, it was by definition

unable to predict locations of all speakers correctly. The reason for this was that the coordinates could be predicted only for an area limited by the class centres' locations, which did not cover the entire country. One could alleviate this issue by shifting the outer centres to the borders of the country. This would allow for a wider range of locations to be predicted. However, this would doubtfully make predictions more correct. Samples which are good representatives of their class would eventually be predicted to correspond to locations at these borders. Instead, it was decided to create networks which would be able to directly predict coordinates without considering any class membership during training.

5.2 A Regression Neural Network for Predicting Coordinates

In order to create a regression neural network which outputs corresponding coordinates for a given i-vector, the network's design had to be changed to include an output layer consisting of just two neurons. These corresponded to latitude and longitude. Consequently, the networks were then trained on i-vectors and latitudes and longitudes corresponding to the participants' locations. The cost function was initially redefined as the absolute difference between actual and predicted coordinates.

5.3 Optimisation

Training was conducted with similar settings to those considered optimal for classification. As before, on-line training was applied throughout 300 epochs. The initial learning rate and its decay were set to 0.001 and 0.99 respectively. Since the regression networks were more prone to over-fitting it was necessary to lower the dropout rate to 0.4. Previously taken measures dealing with different class prior probabilities were, however, not applicable.

The first regression network Model F which was trained in that manner, however, turned out to make poor predictions. As Figure 21 illustrates, it predicted coordinates close to the demographical centre of the country for most of the data. Logically, it performed extremely well when tested on data from speakers who happened to come from that region. On average, the predicted locations were less than 12 kilometres away from the centre. In other words, the model had learned little beyond the average location of all training data.

A closer look at how such a network learns reveals what caused this behaviour. During training, the cost function calculates the difference between the actual coordinates of a sample and those predicted by the model.

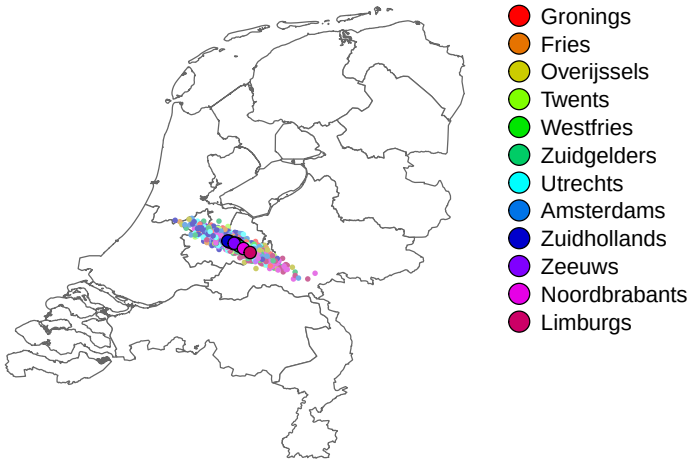


Figure 21: Predicted locations of Model F.

Weights are then adapted such that this difference is reduced. The biggest possible discrepancy between the predicted and actual location of a sample corresponds to the longest distance within the country. However, a network which has learned to return the mean of all coordinates roughly cuts this error in half. Learning the centre coordinates does not require the network to discover any relationships between the i-vectors and their coordinates. This performance could theoretically be achieved even if i-vectors consisted of just random numbers. However, actual learning is required for the network to be able to predict coordinates which are closer to the target. As it turned out the model failed in learning to predict locations beyond the small stretch around the centre. What helped to alleviate this problem was to square the cost.

Further improvements were achieved by stressing those training samples which had previously been shown to be more predictable and lowering the importance of more confusable samples. Each i-vector was assigned a score ranging from 0 to 2 with a higher score indicating better predictability. In training, the cost was multiplied by these scores. Thereby, the network learned more from committed errors on data which were more representative of a certain region. The scores were calculated based on the results of Model E as presented in 5.1. Distances were measured between actual and predicted locations, scaled to range (0-2) and subtracted from 2.[16] Although this approach involved adapting training based on the outcome of a previous

[16] In fact, scaling was separately performed per accent class in order to avoid biasing scores to be higher for locations in the centre.

model this did not constitute a mix-up of training and test data.

5.4 Results

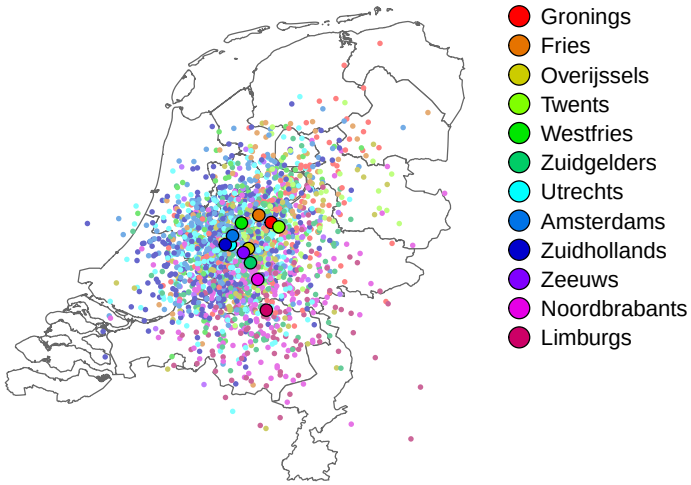


Figure 22: Predicted locations of Model G.

Figure 22 shows locations predicted by Model G, the final regression model. Those are mostly scattered around the centre and largely overlap for all accent regions. However, a look at the mean predicted locations also shows that most of these approach the actual centroids to some degree.

As a measure to evaluate a regression model for locations, van Leeuwen and Orr (2016) propose to use the distance between a predicted location and the actual location. For comparison, the distances between actual locations and the centre[17] were also measured. Taking these two measures allowed to state how the model differed in performance from just making an educated guess purely based on the mean coordinates of all training data.

This also made it possible to assess whether the predicted coordinates merely correspond to locations randomly scattered around the centre or whether they refer to locations closer to the targets. In order to do so, a Wilcoxon signed-rank test was conducted comparing the two measures. The predicted locations were found to be significantly closer to their targets (Mdn = 57.49 km) compared to the distances between centre and targets (Mdn = 60.34 km) , $p < .001$, r = -0.07. Comparing means, predicted locations and targets were 63.40 km apart (SD = 36.59 km) whereas the mean distance

[17]Here, the centre is defined as the average location of all training data.

between centre and targets amounted to 66.67 km (SD = 37.15 km). In other words, the model was able to predict locations which were on average 3.29 km closer to their targets.

However, it should be noted that the reported mean improvement is very conservative. As mentioned above, the number of recordings per region largely differed. Those for which the most data were available were also among those for which the model showed the worst performance (see Table 2). Averaging over the twelve accent regions, the model's prediction were 6.46 km closer to the target than the centre.

A post-hoc analysis was performed consisting of dependent-samples t tests for each of the twelve accent regions. The significance level was lowered from $\alpha = .05$ to $\alpha = .00417$ by applying Bonferroni correction. The analysis showed that the model's predictions were significantly better than random for participants of just five of the twelve accent regions. For *Utrechts*, predictions were found to be significantly worse.

Table 2: *Mean distances in km between predicted and actual locations as well as between the demographic centre and actual locations. Negative differences between these values indicate that the model's predictions were on average closer to the target than the centre. Results of t tests comparing these two measures for participants of each subgroup are shown in the right-hand column. Asterisks indicate significant results for $\alpha = .00417$*

Accent	Distance between Prediction and Target	Distance between Centre and Target	Difference	Post-Hoc t test	p value
Gronings	119.13	139.90	-20.77	$t(145) =$ -6.43,	$< .001$*
Fries	95.24	115.57	-20.33	$t(113) =$ -6.47,	$< .001$*
Overijssels	66.27	59.45	6.83	$t(111) =$ 2.49,	.014
Twents	80.56	95.68	-15.12	$t(95) =$ -6.75,	$< .001$*
Westfries	65.81	72.41	-6.60	$t(88) =$ -2.52,	.014
Zuidgelders	49.38	46.37	3.01	$t(217) =$ 2.15,	.032
Utrechts	33.93	15.85	18.08	$t(300) =$ 15.73,	$< .001$*
Amsterdams	46.67	45.10	1.57	$t(280) =$ 1.17,	.244
Zuidhollands	59.89	61.04	-1.15	$t(384) =$ -0.86,	.388
Zeeuws	116.87	116.27	0.59	$t(68) =$ 0.17,	.866
Noordbrabants	55.32	63.95	-8.63	$t(248) =$ -4.66,	$< .001$*
Limburgs	81.40	116.38	-34.97	$t(130) =$ -14.16,	$< .001$*

6 Comparison with Human Performance

The classification and regression models have so far only been assessed by contrasting their output with random performance. These comparisons suffice to answer the question whether the models actually are able to make any meaningful predictions. However, they are less suitable to assess how well they fare at this task. As mentioned in 2.3.2, location judgements were available for a part of the participants. These were elicited by the smart phone application by playing recordings to other participants and asking them to indicate the speaker's location on a map. In other words, participants had a similar task to what was attempted computationally.

In this section, these human judgements on some of the participants' recordings are compared to corresponding predictions of the classification and regression models. First, it is discussed how human listeners differ from the models in how both deduce someone's accent. This is followed by contrasting human judgements to predictions of the classification model and finally to those of the regression model.

6.1 Human versus Computer Perception of Accents

While participants could make judgements based on the unaltered audio recordings the input used in training and testing the models differed considerably. The i-vectors with which the models were trained were based on down sampled audio data cut into many short frames of just 25 milliseconds. That means any cue about a speakers local accent which could not be expressed within the time frame of these short samples was lost. The duration of vowels and consonants is often sufficiently short to be covered entirely by such a sample. Syllables, however, will most often be too long. Consequently, the models should be expected to be blind to differences in phonotactics and prosody. Human listeners, however, can also make use of all available cues beyond the phone level. That, for instance, includes observing differences in consonant clusters. The participants may have noticed differences with respect to the frequency of schwa insertion, which is a phenomenon more common in the south of the country (Swerts, Kloots, Gillis, & De Schutter, 2001). They may also have noticed differences in intonation and theoretically even have exploited lexical differences when trying to guess a speakers location. Although, such cues were expected to be scarce given the relative shortness of the recordings. Another advantage for human listeners is that they must have been exposed to far more speech throughout their lives than what has been recorded for this project. Having access to the full amount of information of a recording as well as being experienced with different ac-

cents potentially enables humans to make better predictions than the models presented below. However, most of the participants can be expected to not have been explicitly trained to recognise local accents. Their exposure to different accents is unlikely to have been even. At its worst, however, participants should still have been good at detecting whether someones accent was similar or different to their own.

In sum, the question is whether one should expect participants to outperform the models presented here because recordings were made available in their original form including all cues about the speakers' locations or whether models explicitly trained for accent location show a better performance.

Previously, it has been shown that it is possible to train models which can outperform humans when given the tasks of language accent classification. For instance, Arslan and Hansen (1996) used Hidden Markov Models to model sequences of phonemes of (foreign) accented and unaccented American English. Comparing their performance to human listeners, the model correctly classified 68.8% whereas human listeners on average were able to correctly classify just 52% of the used word list. The authors point out individual differences in listeners' performance. Of those, two were still slightly better at this task than the model. Model and listener performance was reported to be similar when given the task of deciding whether a word was accented or not, i.e., accent detection. In a more recent dissertation by Hanani (2012), different models are compared to human listeners in how well they can classify British accents. The most basic acoustic model (GMM-UBM) performed just slightly better than human listeners given samples of 30-45s length. Model and listeners correctly classified 60.2% and 58.24% respectively. However, when phone sequences were modelled performance improved by more than 20%. From these results, it can be concluded that the models presented above should perform at least as good as humans provided that there are no technical issues.

A fair comparison requires that human judgements and the model's predictions refer to recordings of the same people. Therefore, the output of the final models is presented again showing only predictions for those 897 speakers whose locations were also guessed by other participants. However, it cannot be guaranteed that exactly the same set of recordings was used. Human judgements were considered only for recordings deemed the most informative. There should be a large overlap with recordings used in prediction. However, some bias towards better human performance can be expected (see also: footnote 5 in 2.3.2).

6.2 Classification

Van Leeuwen and Orr (2016) argue that human perception of accents is more comparable to a classification task. However, participants were not explicitly asked to classify accents of fellow participants (e.g by choosing a label from a list of accents). Instead, they were asked to guess corresponding locations which were then registered as coordinates. In order to compare these with the output of the classification model, coordinates had to be matched with corresponding accent labels. For participants whose location was guessed more often than once by the same listener, mean coordinates were calculated.

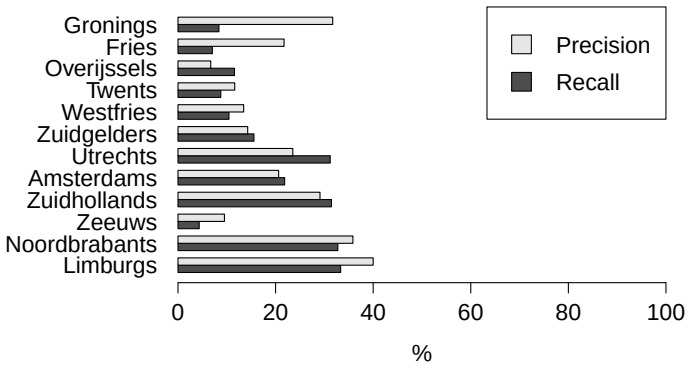
Table 3: *Confusion matrix for classifications based on guessed locations of fellow participants. Rows indicate the accents to which the recordings actually correspond. Columns indicate the accent as which a recording was classified.*

	G	F	O	T	WF	ZG	U	A	ZH	Z	NB	L
Gronings (G)	**13**	5	19	8	3	13	34	24	28	1	5	2
Fries (F)	5	**5**	11	2	1	7	16	6	12	1	5	0
Overijssels (O)	1	0	**11**	6	5	4	28	21	14	0	5	0
Twents (T)	3	2	8	**5**	2	5	13	7	4	0	7	1
Westfries (WF)	1	0	9	0	**7**	2	12	19	14	1	2	0
Zuidgelders (ZG)	1	2	12	5	4	**26**	32	21	25	2	31	6
Utrechts (U)	7	3	23	2	9	30	**112**	62	85	2	22	2
Amsterdams (A)	6	3	29	5	14	23	82	**66**	57	3	12	2
Zuidhollands (ZH)	3	0	29	6	5	24	97	66	**117**	7	15	3
Zeeuws (Z)	0	1	4	0	1	6	8	5	9	**2**	8	2
Noordbrabants (NB)	0	2	8	2	1	31	34	21	34	2	**76**	21
Limburgs (L)	1	0	1	2	0	11	8	2	3	0	24	**26**

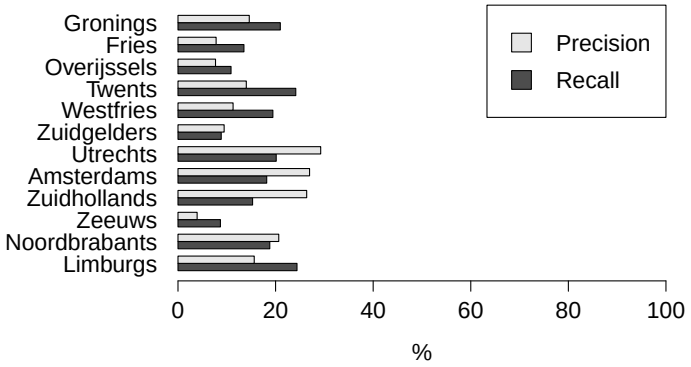
Overall, the participant's location judgements are equivalent to a mean recall of 18.06% and a precision of 21.51%. With 16.94%, Model E reached a somewhat lower mean recall when tested on recordings of the same participants. The mean precision of 15.64%, however, is considerably lower in comparison to human performance.

Human guesses differed from the model's predictions in that there was a clear preference for those locations with the highest population densities. What is striking is that more than half of all participants were guessed to have either an *Utrechts*, *Amsterdams* or *Zuidhollands* accent. At the same time, speakers were rarely considered to come from regions corresponding to *Gronings*, *Fries*, *Twents* or *Zeeuws*. This pattern may reflect the participants' knowledge about Dutch demographics. In other words, participants may have based their guesses on both the perceived accent as well as the prior

probabilities for speakers to stem from a different regions. The classification model, however, was explicitly trained such that it would not include different class prior probabilities in prediction.



(a) Precision and recall based on guesses by fellow participants.



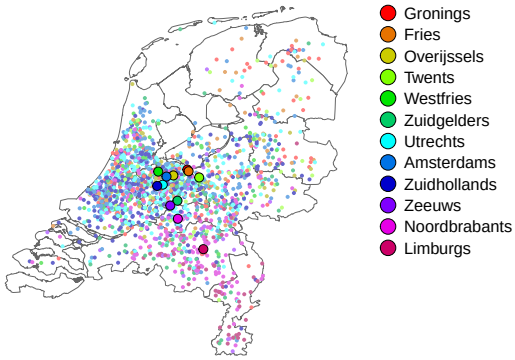
(b) Precision and recall of Model E tested on the same subset of recordings.

Figure 23: Precision and recall for accents guessed by fellow participants and predicted by Model E.

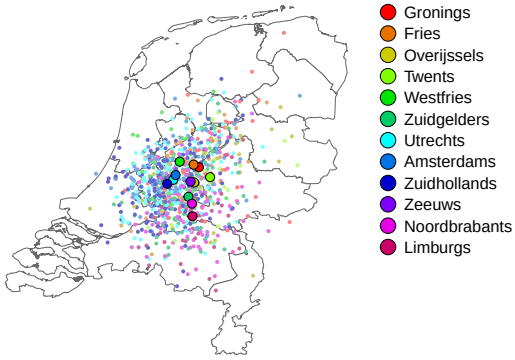
In summary, the participants and the classification Model were similarly likely to match a recording with the correct class. However, given a number of recordings which all had been classified as the same accent, the proportion of those to be correct was on average higher for participant's guesses.

6.3 Regression

The participants' task of guessing a fellow participant's location can be more directly compared with the regression model. This is also considered more suitable since it is unknown whether the results as presented above would have been the same if instead of pointing at a location the participants had been asked to select a suitable accent from a list. By allowing participants to freely point at any location on the map, they were not forced to select just one region. If they considered a fellow participant's accent to show features of different regional accents, they could also select the point in between those regions.



(a) Locations as guessed by fellow participants.



(b) Predicted locations by Model G.

Figure 24: Participants' locations according to fellow participants' guesses and predictions of regression Model G

Figure 24 shows guessed locations as well as Model G's predicted locations for the same group of participants. Guessed locations were visibly more spread than predicted ones. The former were on average placed 49.14 km

off the centre whereas the latter were on average much closer with 34.15 km. However, both were similar in that predictions and guesses largely overlapped with regard to the speakers' accent regions.

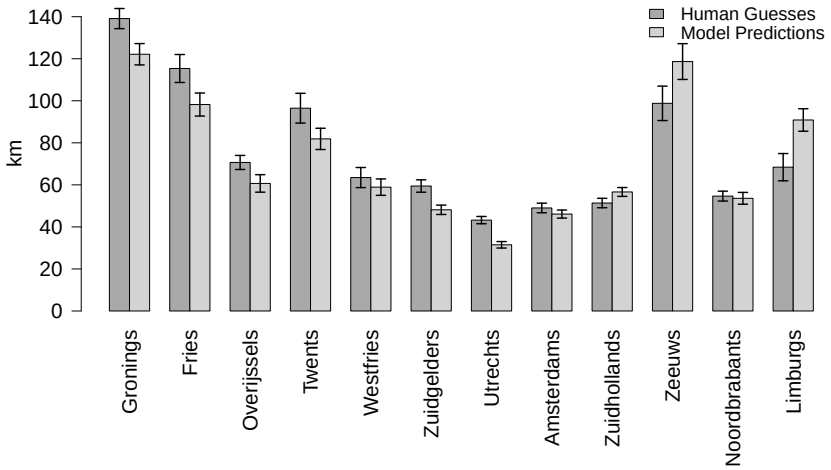


Figure 25: Mean distances between speaker's actual locations and their locations according to guesses by fellow participants and predictions of Model G. Error bars indicate ± SEM

A quite similar pattern can be observed considering the distances of both guesses and predictions to the actual locations of the speakers. As Figure 25 shows, guessed and predicted locations were closer to their targets if these were located near the centre of the country. Furthermore, humans appear to outperform the model when guessing the location of speakers of *Zuidhollands*, *Zeeuws* and *Limburgs*. However, the model's predictions were on average closer or equally close to their targets with regard to speakers from the remaining accent regions. In other words, differences between human guesses and the model's predictions seem to depend on the accent region to which speakers belonged. This was tested by performing a factorial ANOVA. However, instead of taking the absolute distance between target and prediction (or guess) as a measure of performance, only improvement over taking the location of the demographic centre as prediction for all speakers (a measure introduced in 5.4) was used. That is, the ANOVA was conducted to compare the main effects of prediction type (model prediction, human guess) and accent region as well as the interaction of these two on improvement over predicting the centre location. The main effect of prediction type on improvement was significant, $F(1,2843) = 4.77$,$p = .029$. There was also a

significant main effect of accent region on improvement, $F(11, 2843) = 47.52$, $p < .001$. The interaction effect between prediction type and accent region was significant as well, $F(11, 2843) = 4.11$, $p < .001$. This suggests that the improvements regarding speakers of different accent regions were differently affected by whether predictions were made by the model or a fellow participant.

Table 4: *Estimated coefficients of the linear model. Colons are used for interaction terms.*

	Estimate	Std. Error	t value	p value	
(Intercept)	-8.04	1.92	-4.18	<.001	***
pred. MODEL	7.56	3.46	2.18	.029	*
Fries	10.52	4.37	2.41	.016	*
Gronings	16.76	3.29	5.09	<.001	***
Limburgs	59.56	4.25	14.01	<.001	***
Noordbrabants	18.35	2.91	6.31	<.001	***
Overijssels	-6.50	3.90	-1.67	.096	
Twents	12.22	4.82	2.54	.011	*
Utrechts	-19.60	2.60	-7.52	<.001	***
Westfries	11.18	4.53	2.47	.014	*
Zeeuws	16.28	5.24	3.11	.002	**
Zuidgelders	-2.76	3.23	-0.86	.392	
Zuidhollands	13.56	2.59	5.24	<.001	***
pred. MODEL:Fries	6.48	7.55	0.86	.391	
pred. MODEL:Gronings	-0.39	6.06	-0.06	.949	
pred. MODEL:Limburgs	-32.61	7.28	-4.48	<.001	***
pred. MODEL:Noordbrabants	-7.86	5.25	-1.50	.135	
pred. MODEL:Overijssels	5.18	6.88	0.75	.451	
pred. MODEL:Twents	5.23	8.32	0.63	.529	
pred. MODEL:Utrechts	4.57	4.71	0.97	.332	
pred. MODEL:Westfries	0.59	7.69	0.08	.939	
pred. MODEL:Zeeuws	-23.61	9.13	-2.59	.010	**
pred. MODEL:Zuidgelders	0.15	5.70	0.03	.978	
pred. MODEL:Zuidhollands	-9.23	4.68	-1.97	.049	*

Table 4 shows the corresponding contrasts. It confirms above made observations that overall model's predictions outperform human guesses with the exception of *Zuidhollands*, *Zeeuws* and *Limburgs*. For these, significantly smaller improvements are attested if predictions stem from the model.

7 Discussion

Although both versions of the location system have been shown to be able to make meaningful predictions, their performance can only be described as underwhelming. The classification model turned out to perform only about ten percent better than chance. When attempting to predict people's exact locations, the regression model's predictions were just about six kilometres closer to their targets compared to just guessing every speaker to come from the country's centre. There are two potential causes for this performance considered here. The system's implementation may simply be deficient. On the other hand, the available data may be insufficient or not suitable for the task. Both explanations are not mutually exclusive. In other words, the system's performance may reflect problems regarding both the implementation and the used data.

Although the final models are the best out of several hundred experimental versions there is no guarantee that no further improvement would be possible. In fact, it is considered extremely likely that better solutions could be created given the virtually infinite possibilities of implementation. For instance, there was no discussion of different network architectures above. All of the above models were identical with respect to their input and hidden layers. Some alternative architectures had been tested beforehand, however, it is considered likely that more suitable solutions could be found through further experimentation. Furthermore, as discussed in detail in section 4.3, adapting parameters in training models is crucial and often has big effects on their eventual performance. Despite having extensively experimented with different settings, there may still be some improvement possible by further fine-tuning training parameters. Another way to improve performance would be to make the models' input even more information rich. This can, for instance, be achieved by using features which cover more cues of local accents such as differences regarding phonotactics and prosody.

Comparing the models' performance in locating accents with that of human listeners puts the results somewhat in perspective. It was expected that the models should be outperformed by human listeners if the implementation was severely flawed. This was not found to be the case. Main differences in accuracy between a model's predictions and human guesses were found to depend on specific accent regions. When comparing human guesses with predicted locations of the regression model, human listeners were able to outperform the model with respect to recordings of just three accent regions. Possible explanations for the model's worse performance in these cases may differ with respect to the three regions. Of those, the biggest difference was found regarding speakers from the accent region labelled as *Limburgs*. In the

literature, several dialects are grouped together under this name. Of those many (however, not all) have a pitch accent (Gussenhoven & Peters, 2008). This attribute, which contrasts with other Dutch accents, may have helped human listeners in identifying speakers of this region. However, the model could not exploit this difference as it was not covered by the used features.

With regard to *Zeeuws*, the model's performance may be attributed to the fact that the available data were simply scarcest for the corresponding region. For *Zuidhollands*, human guesses were just slightly better than the model's predictions. As has been discussed above, it appears that listeners defaulted to this and the two neighbouring accent regions *Utrechts* and *Amsterdams* in guessing speakers' locations.

The similarly poor performance of human listeners in guessing locations of participants indicates that the used data were problematic. In section 2, it was pointed out that they were not representative of the Dutch population. Moreover, it was apprehended that speakers of standard Dutch would predominate. Consequently, the data would include fewer recordings of local accents. A principle components analysis on the extracted i-vectors as presented in section 3.3 showed that just little of the covered variability can be attributed to local differences.

Apart from qualitative issues of the data affecting both human and model performance, the data are also problematic in terms of quantity with respect to some regions. As has been pointed out above, many of the participants came from highly populated areas of the Netherlands, whereas there were just few from northern regions and the south-west. Consequently, some accents had to be abstracted from very few recordings in training the models.

Obviously, alleviating the problems described above would involve collecting more data. In the meantime, the *Sprekend Nederland* project has repeatedly been covered on Dutch television. Previously, the project had exclusively been featured on scientific programmes. However, different audiences are expected to have been reached by also receiving airtime on *RTL Late Night* as well as broadcasting a 90-minute game show carrying the project's name[18]. At the time of writing, more participants have been recruited and more data have been collected via the smart phone application. Having more recordings per accent region would allow for future models to make better generalisations. Furthermore, more meta-data and judgements would help with filtering out recordings of speech without local accents. There is a good

[18]The corresponding clip from *RTL Late Night* can be found on the broadcaster's website: `http://www.rtlxl.nl/#!/programma-301978/a2c480c8-e789-4413-a1de-d842079fc8c8`

The entire *Sprekend Nederland* show can be watched on the NPO's website: `https://www.npo.nl/sprekend-nederland/19-05-2016/VPWON_1260835`

chance a better performance could be achieved by retraining the models on these new data.

8 Conclusion

In this thesis, two versions of an accent location system were introduced. The first version was built to classify recordings of participants as belonging to one out of twelve accent regions, whereas the second system predicted the participants' coordinates. Both were built using data collected via a smart phone application as part of the project *Sprekend Nederland*. Data included recordings of speech, judgements about other participants' speech and meta-data including the participants' locations. By the time work on the accent location began, these data were mostly incomplete. Major challenges faced during engineering the systems were due to dealing with an unbalanced dataset which was especially dominated by young participants from the country's metropolises. Furthermore, participants' self-reporting, judgements of fellow participants, and the results of a principal component analysis of the used features, all indicated that the recordings would contain just little speech with local accents. Building an accent location system on these data consequently required exploiting scarce cues of these accents. It was chosen to implement both versions of the system as neural networks, which proved to be sufficiently flexible for the given task allowing for predictions above chance level. In general terms, their performance was still poor. However, when comparing the system's ability to predict a speaker's location to that of human listeners, both turned out to be overall similar. This can be interpreted to indicate that most of the available information had been exploited in training the system. While acknowledging the technical possibilities for improvement, the main causes for its poor performance are considered to be the problems with data as described above. Nevertheless, this thesis illustrated how the data collected by *Sprekend Nederland* can be used for accent location. As the collection continues to grow, better performances are expected to be achieved by reimplementing the system using an updated dataset.

Acknowledgements

Throughout my master's, I could count on the support of many people and I would like to take the opportunity to thank them.

First of all, I like to express my gratitude to Rosemary Orr, who was not just a fun person to work with but also someone who would always encourage others to venture forward and try out new things. It was Rosemary who first challenged me to take the *Sprekend Nederland* data and implement an accent location system.

I was fortunate enough to have had two supervisors for my thesis project, David van Leeuwen and Gerrit Bloothooft. Without David, this work would not have been possible. He was incredibly helpful by organising the logistics of the project, regularly sharing his desk with me and helping me through most of the problems I encountered with machine learning. I am also grateful for Gerrit's excellent supervision, for his valuable advice, for pointing out the gaps in my drafts and generally showing lots of patience.

Much of the thesis work has been carried out in Nijmegen. I very much enjoyed the buzz at NovoLanguage and wish them all lots of success for the future. Thanks also to Radboud University's Centre for Language Studies for granting access to their cluster.

Furthermore, I would like to thank everyone with whom I also had the pleasure to work throughout the last few years: Hugo Quené, Jacky-Zoë de Rode, Aoju Chen, Frans Adriaans and Alexis Dimitriadis.

Having mainly a background in childhood education, moving on to linguistics did not appear to be the most logical step for many. I especially would like to thank Angelika Bonczyk and Tamar Keren-Portnoy for actively supporting that decision.

I had a very smooth start in Utrecht thanks to Damar Hoogland, Marijke Struijk, Anna Bruggeman and their families who had prepared me for all the Dutchness that was to come.

For most of my degree, I remember sitting in the Janskerkhof basement which was not as terrible as it might sound thanks to all my classmates who shared the same fate. I would like to thank all of them for the mutual support and especially all the laughter. Special thanks to Shuangshuang Hu, Marleen Berkhout, Joe Rodd and Juanmi Vicente Flores. Let me also thank my friends and passionate Dutch teachers Jorik van Engeland and Erlinde 'Langlaufen' Meertens.

Last but not least, I want to thank my entire family for their endless support.

Bedankt allemaal!

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., ... Devin, M. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Arslan, L. M., & Hansen, J. H. (1996). Language accent classification in American English. *Speech Communication*, *18*(4), 353–367.

Bahari, M. H., Saeidi, R., & van Leeuwen, D. (2013). Accent recognition using i-vector, gaussian mean supervector and gaussian posterior probability supervector for spontaneous telephone speech. In *Proceedings ICASSP'2013* (pp. 7344–7348).

Bates, D. M. (2010). lme4: Mixed-effects modeling with R. *URL http://lme4. r-forge. r-project. org/book*.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architectures. In *Neural Networks: Tricks of the Trade* (pp. 437–478). Springer.

Biadsy, F. (2011). *Automatic dialect and accent recognition and its application to speech recognition* (Unpublished doctoral dissertation). Columbia University.

Centraal Bureau voor de Statistiek. (2016a). *CBS StatLine - Bevolking; geslacht, leeftijd en burgerlijke staat, 1 januari* [Data set]. Retrieved 2016-12-15, from `http://statline.cbs.nl/Statweb/publication/?DM=SLNL&PA=7461BEV&D1=0&D2=a&D3=0,19-133&D4=0,65-66&HDR=G3,T&STB=G1,G2&VW=T`

Centraal Bureau voor de Statistiek. (2016b). *CBS StatLine - Bevolking; hoogstbehaald onderwijsniveau en onderwijsrichting* [Data set]. Retrieved 2016-12-15, from `http://statline.cbs.nl/Statweb/publication/?DM=SLNL&PA=82816ned&D1=0&D2=0&D3=0&D4=0&D5=0,3-4,8-10,l&D6=0&D7=64&HDR=G3,G2,G1,G6&STB=G5,T,G4&CHARTTYPE=1&VW=T`

Centraal Bureau voor de Statistiek. (2016c). *CBS StatLine - Bevolkingsontwikkeling; regio per maand* [Data set]. Retrieved 2016-12-15, from `http://statline.cbs.nl/Statweb/publication/?DM=SLNL&PA=37230ned&D1=0,4-5,7-8,13-16,20&D2=0-1,5-16&D3=8-9,47-48,112-113,177-178,181,190-191&HDR=G2&STB=G1,T&VW=T`

Chambers, J. K. (1992). Dialect acquisition. *Language*, *68*, 673–705.

Daan, J., & Blok, D. P. (1969). *Van randstad tot landrand: toelichting bij de kaart Dialecten en naamkunde*. Amsterdam: Noord-Hollandsche uitgevers maatschappij.

Glembek, O., Burget, L., & Matejka, P. (2015). *Voice Biometry Standard-Draft.* Retrieved from `http://voicebiometry.org/`

Gussenhoven, C., & Peters, J. (2008). De tonen van het Limburgs. *Nederlandse taalkunde*, *13*, 87–114.

Hanani, A. (2012). *Human and computer recognition of regional accents and ethnic groups from British English speech* (Unpublished doctoral dissertation). The University of Birmingham, Birmingham.

Heeringa, W. J. (2004). *Measuring dialect pronunciation differences using Levenshtein distance* (Unpublished doctoral dissertation). University of Groningen, Groningen.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Jessen, M. (2007). Speaker classification in forensic phonetics and acoustics. In *Speaker classification I* (pp. 180–204). Springer.

Kaart package. (2013). Retrieved from `http://www.meertens.knaw.nl/kaart/downloads.html` (version 3.0.1)

Kuznetsova, A., Brockhoff, P. B., & Christensen, R. H. B. (2015). *lmerTest: Tests in Linear Mixed Effects Models. R package version 2.029. 2015.*

Lawrence, S., Burns, I., Back, A., Tsoi, A. C., & Giles, C. L. (1998). Neural network classification and prior class probabilities. In *Neural networks: tricks of the trade* (pp. 299–313). Springer.

Myers, G. (2006). 'Where are you from?': Identifying place. *Journal of Sociolinguistics*, *10*(3), 320–343.

Orr, R., Quené, H., van Beek, R., Diefenbach, T., van Leeuwen, D. A., & Huijbregts, M. (2011). An International English Speech Corpus for Longitudinal Study of Accent Development. In *INTERSPEECH* (pp. 1889–1892).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*(Oct), 2825–2830.

R Core Team. (2015). *R: A language and environment for statistical computing.* Vienna, Austria. Retrieved from `https://www.R-project.org/`

Smakman, D. (2006). *Standard Dutch in the Netherlands: A sociolinguistic and phonetic description* (Unpublished doctoral dissertation). Netherlands Graduate School of Linguistics, Utrecht.

Swerts, M., Kloots, H., Gillis, S., & De Schutter, G. (2001). Factors affecting schwa-insertion in final consonant clusters in standard dutch. In *INTERSPEECH* (pp. 75–78).

Tennekes, M. (2016). *tmap: Thematic maps.* Retrieved from `http://CRAN.R-project.org/package=tmap` (R package version 1.6-1)

van Leeuwen, D. A., & Orr, R. (2016). The "Sprekend Neder-

land" project and its application to accent location. *arXiv preprint arXiv:1602.02499*.

Walt, S. v. d., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, *13*(2), 22–30.

Wilson, D. R., & Martinez, T. R. (2003). The general inefficiency of batch training for gradient descent learning. *Neural Networks*, *16*(10), 1429–1451.

Woodland, P. C., Odell, J. J., Valtchev, V., & Young, S. J. (1994). Large vocabulary continuous speech recognition using HTK. In *Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on* (Vol. 2, pp. II/125–II/128 vol. 2). Ieee.