Radboud University Nijmegen

Faculty of Science

Automatically detecting head-shakes in NGT conversations

Thesis MSc Data Science

Supervisor:
Martha Larson

Author:
Cas van Rijbroek

Second reader:
Tom Heskes

September 2023

Contents

Abstract

Non-manuals play an important role in sign language. For this research, the role of head-shakes in Dutch sign language was investigated by creating a detection system based on pre-trained models and Hidden Markov models. The system yielded a precision of 0.1 and recall of 0.6 on the simplest evaluation method. We propose new evaluation techniques and use the detection system to assist linguists in their research. Setting up the ground work for future research into this area.

1 Introduction

Sign language functions like spoken language. They consist of phonetic components that can be combined in many ways to construct sentences. In a spoken language, these components relate to the uttering of sounds, while for sign language, these refer to the production of manual signs (hand location, movement, shape and orientation) and non-manual features such as head movements and facial expressions. These non-manual features not only accompany signs, they can change the meaning of a sign entirely. In this research we are interested in the function of non-manuals in Dutch sign language.

This work places a focus on the role of head-shakes in Dutch sign language (NGT). We focus on head-shakes for two reasons. Firstly, since non-manuals in sign language are a relatively understudied field of research, we intended to lay the groundwork for future research into different kinds of non-manuals by narrowing down on a specific kind first, allowing us to research it more in depth. Secondly, head-shakes are a relatively common expression, both in sign language and spoken languages. By choosing head-shakes we were able to utilize a relatively large set of existing annotations, which were not readily available for other non-manuals.

For this work we made use of an annotated dataset of video recordings made by researchers from Radboud University containing dialogues between NGT signers. This dataset contains partial annotations for translations on sentence and sign level, it also includes annotations for head movements. Because the overwhelming majority of these annotations describe head-shakes in different contexts, this work will focus on head-shakes by developing a predictive model that can detect head-shakes from video data. The goal of this work is to produce a model that can be useful to linguists in their research. Specifically, we used pose estimation to train Hidden Markov models to predict head-shakes in video data of NGT.

NGT is the language of the deaf community in the Netherlands. While the number of deaf people in the Netherlands is not registered officially, using international trends it is estimated at around 11,900-20,400 people [1]. Accounting for the siblings and parents of these people, it is further estimated that around 60.000 people use NGT [2]. At the time of this estimation there were around 17,000,000 people in the Netherlands [3], putting the NGT community at around 0.35% of the Dutch population.

To protect this minority, the Dutch government has decided to acknowledge NGT as an official language, this codifies rights of the deaf community in the Netherlands. For example, they may now swear an oath of office using NGT. Additionally, any communications to the public regarding a crisis must be translated to NGT [4]. Despite these efforts, the body of linguistic research into NGT is limited.

Specifically, there is currently no work that investigates the role of non-manuals in NGT using a data-driven approach. With this thesis we wanted to fill that knowledge gap by making the first contribution to this line of research, making use of approaches inspired by literature.

The field of machine learning has gained much traction over the past decade. We define this as any task solved with (the assistance of) a machine which would otherwise be too challenging for a human to complete, in particular we refer to the process which determines the parameters for models designed to solve such tasks. Deep learning is a sub field of machine learning which tackles these problems with particularly large models, typically consisting of many layers (hence deep learning). With the rise of increasingly complex deep learning solutions, it has become difficult for humans to understand how models come to a decision. For this reason the field of XAI has gained traction over the last few years [5]. XAI involves the development of methods that can explain and interpret the output of machine learning models. Because this work seeks to facilitate future research into non-manuals in sign language, we wanted to pay special attention to making our solutions explainable. We chose to work with explainable statistical models which make use of meaningful features, rather than immediately applying deep learning algorithms to the problem. In addition, we performed an in-depth failure analysis of the predictions made by our best algorithm to gain more understanding of the limits of the methodology and the dataset itself.

1.1 Related work

1.1.1 Non-manuals and negation

Investigation of the role of non-manuals in sign language concludes that they are essential to the grammar of these languages [6]. In fact, it is argued that if you were to obscured the face of a signer in a recording, a good part of the message would be lost. Non-manuals can create lexical distinctions between signs and modify the meaning of a sentence entirely. There exists large variation in the role of specific non-manuals between sign languages. In spite of this, head-shakes are commonly found to be involved with negation across sign languages. A distinction can be made between languages that are manual dominant, where negation still requires a manual component, and non-manual dominant, where negation can be achieved by the head-shake alone. By developing tools to automatically analyse NGT video footage for these non-manuals, we can facilitate future research into this direction.

There has been a typology study of NGT that classifies the language as non-manual dominant [7], however, as [8] shows, such typological classifications are not necessarily accurate. By studying this aspect of the language using a data-driven approach, we can gain new insights into the grammar of NGT and place that research in international context.

Internationally there exists a significant body of research into the function of non-manuals in sign language. Johnston has done a study of the role of non-manuals (head-shakes) in negation for Australian sign language (Auslan) [9]. He found that head-shakes accompany negation in roughly half of the manually negated clauses, but the negation rarely depends on the head-shake. This classifies Auslan as a primarily manual dominant language, but Johnston questions this typology. The findings of his study and a previous analysis of Kata Kolok did not determine that head-shakes have been incorporated in these language's grammar. but Johnston points out that there is a lack of data in this regard for other languages. While these works do not directly address NGT, insights into the workings of other sign languages can help us place what we learn in a larger context. Additionally, we can use effective research methods from this international body of work and apply them to NGT.

1.1.2 Data-driven approaches

Chizhikova and Kimmelman took a data-driven approach to the analysis of head-shake negation in Russian sign language (RSL) [8]. They used face pose estimation software to extract head movements from head-shakes in a corpus of mostly monologues. They found that RSL contains even fewer head-shake markers for negation: a head-shake is not enough to negate an utterance and less than 30% of the negative structures contained a head-shake. This work is an example of how data-driven work can help us test existing hypotheses about sign language.

1.1.3 Sign language recognition

Sign language recognition (SLR) is a relatively popular field of study. It concerns itself with the detection and classification of signs from any sign language in video footage. Some of these works use non-manuals as additional markers for particular signs. For example, [10] did a survey of existing work in SLR for Indian Sign Language (ISL). They found that most work focuses entirely on manual signs, a handful of papers incorporated a non-manual aspect: position of the eyes relative to the hand, head movement and emotions expressed by the face. For example, [11] created a proof-of-concept sign language tutor system, as part of this system they trained a HMM to do SLR based on manual and non-manual features. These features consisted of head movements and facial expressions, they found that incorporating these features into the classifier significantly increased performance compared to only using the manual features.

Brock, Farag and Nakadai [12] developed a sign language recognition system, including various non-manual markers by pose estimations and binary random forests to segment signer activity. These segmentations were then classified for non-manual markers using a CNN. They found that they were able to distinguish segments of non-manuals with 86% accuracy, and were able to improve their word error rate by 13.22% compared to using labels insensitive to non-manuals. This work further corroborates the importance of non-manuals in sign language.

1.1.4 Explainable AI

For this research, we want to ensure that the models we create will be useful to linguists, XAI can help us verify this by seeing whether our models focus on the same features a human would when examining these videos instead of spurious features. Explainability is also partly inherent to the methods we chose for this research. Because this research lays the groundwork for head-shake detection systems in NGT, and because the amount of available annotated data is relatively small for the requirements of deep learning systems, we chose to work with pre-trained models for pose estimation and statistical models that do not rely on learning deep feature representations.

1.1.5 Pose estimation

Pose estimation involves prediction the position of humans from image or video data by finding the exact position of different keypoints, spanning from the eyes to the feet [13]. The exact location and number of keypoints depends on the implementation. Figure 1 shows an example the output of a pose estimation system.

1.1.6 Hidden Markov models

Hidden Markov models (HMMs) are one of the most popular statistical models that model an observable Markov process, this means that any part of the sequence is independent of all parts that came before it, given the previous part. In addition to this, HMMs assume that there exists a hidden process (hence the name) that governs the observable process. HMMs are popular in many fields of study, from microbiology [14] to speech recognition [15] and indeed sign language recognition [16]. In essence, a HMM models probabilities of transitioning to different states given the previous state. You can explicitly train the hidden process to represent certain target labels [14] or you can calculate the probability of observing a sequence given the parameters of the model [17]. Either way, the parameters of a HMM can either be manually set or inferred from data using the Baum-Welch algorithm, which is an expectation-maximization algorithm that determines the optimal transmission (observable process) and emission (hidden process)

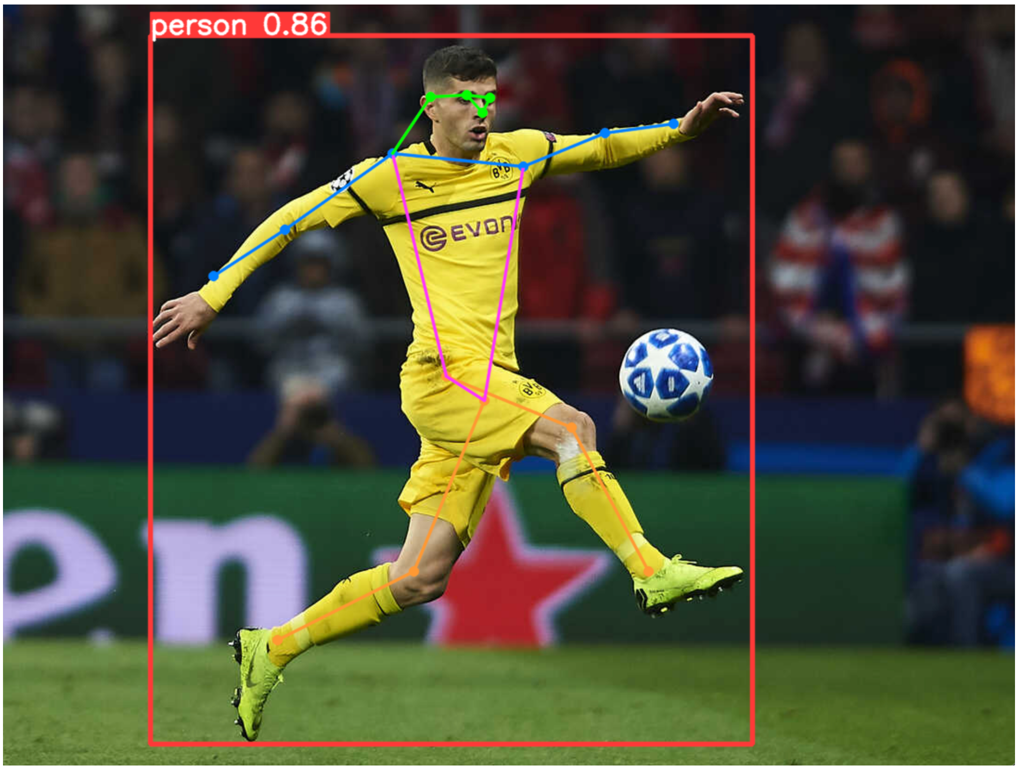

Figure 1: An example of pose detection performed by YOLOv8 (the model used in this work). It outputs a bounding box and keypoint coordinates, each of which have their own confidence values attached. Source: https://github.com/ultralytics/ultralytics/issues/2184

probabilities by iterating over a dataset of sequences, processing them in forward and reverse order.

1.1.7 Evaluation metrics

During this work we have developed several predictive models to detect head-shakes in NGT conversation footage. Since we have access to ground truth annotations of these head-shake events, this is considered a supervised learning problem. In supervised learning, it is common practice to make use of quantitative evaluation metrics [18], typically computed using the number of true positive, false positive, true negative and false negative hits the model made on the target labels. For this work we will evaluate our models using a range of the most popular performance metrics in supervised learning:

1. Accuracy: this metric gives a general performance indication of the system by taking the fraction of correctly predicted instances (positive and negative) in the dataset. It is a simple and intuitive way to measure performance.

2. Precision: as the name implies, this metric measures how precise the predictions of the system are. In other words, how many of the positive predictions made by the system correspond to the correct ground truth label?

3. Recall: this metric measures the fraction of positive ground truths that have been identified by the system.

4. F1-score: the F-score is a metric designed to balance the precision and recall of a system. It depends on a parameter β that determines the importance of recall compared to precision. When $\beta = 1$, we describe the F1-score, which is the harmonic mean between precision and recall.

In addition to this, we will make use of an alignment method to calculate similarity between all annotations on a single video fragment and the ground truth annotations corresponding to that video. Alignment methods try to match two strings by adding, removing and substituting the least amount of characters as possible. For this work, the strings are the predicted labels and the ground truth annotations of a video fragment. The number of changes determines the metric outcome (less changes means a better score).

2 Research Questions

This work aimed to take a data-driven approach to the analysis of non-manuals in NGT, with a focus on explainable AI. We developed several baseline and a HMM based approach to the task of automatically detecting head-shakes in NGT conversation footage. Specifically, this work answers the following research questions:

2.1 Research question 1

Can we automatically detect head-shakes from NGT conversation video footage using by leveraging pre-trained models?

This question is aimed at the technical part of the work, are we able to solve the task of head-shake detection using the proposed methods? We also investigate to what extent the performance of the system is better or worse than random and simple methods. This also involves comparing the results to what we have seen in literature.

2.2 Research question 2

Do head-shake detection models trained on NGT data focus on the same features a human would focus on?

This question was answered through the failure analysis of the model, rather than just the evaluation metrics. We sought to find out if the predictions of the model match human intuition and if so, to what degree?

2.3 Research question 3

Are head-shake detection models sensitive to signers in NGT?

This question addresses another angle of the quality of the model and possibly the dataset itself. Are there any biases in the model performance as a result of the identity of the signer? This question was partially addressed by the failure analysis itself, but also through a statistical significance test on the results aggregated by (anonymized) signer identity.

2.4 Research question 4

How can head-shake detection systems be used to facilitate linguistic research?

An important aspect of any piece of work is impact. For this work we wanted to create something that could be used to facilitate linguistic research in the future. As such, we have conducted an experiment with help of a domain expert to provide an example of how a system such as the one developed during this work can be useful to linguists.

3 Approach

3.1 Head-shake detection

For the first part of the thesis (RQ 1), the goal was to develop a system that can detect head-shakes in NGT video footage. We have detected head-shakes from a dataset using a combination of extracted pose information and Hidden Markov models inspired by head-shake detection literature, leveraging pre-trained models for pose estimation. A similar technique to detect head-shakes has been successfully implemented in the past for systems unrelated to sign language. For example, [19] identifies head-shakes and head nods for human affect recognition (identification of human emotions) and [17] describes a similar technique for detecting head-shakes and head nods. Both techniques use pose estimation combined with HMM's to make predictions in real-time. These papers report high detection rates on their datasets.

Since this is the first work investigating head-shakes in NGT video footage using an automated approach, we required a baseline to compare the performance of the system to. We have developed a random baseline model to establishes a lower bound for this task and a memory-based system that compares the distribution of head movement directions.

3.2 Evaluation metrics

To get a complete picture of how the models developed during this work perform, the problem was evaluated using different metrics. We used accuracy, precision, recall and the F1-score as evaluation metrics, these are introduced with more detail in section 1.1.7.

Because naively calculating metrics over the frame-level predictions gives an incomplete picture of the quality of detection systems, multiple evaluation approaches were implemented and compared for this work. These can be separated into three categories:

1. Frame-level evaluations: performance metrics calculated by treating every frame as an individual label, rewarding/punishing predictions by their exact correspondence to the ground truth. On this level we calculated accuracy, precision, recall and F1 score to get a balanced overview of the performance of the system.
2. Sequence alignment evaluations: measuring prediction quality by the number of changes that have to be performed on the prediction sequence to align it with the ground truth. For this metric we chose the normalized Levenshtein distance, which is a string metric that calculates a cost for the number of insertions, deletions and substitutions required to match two strings. For the implementation the frame-level predictions and ground truth were treated as strings.
3. Event-level evaluations: performance metrics calculated by treating ground truth annotations as single targets. To achieve this, connected frame-level annotations and predictions of the same class were treated as single events, recording their starting and ending times in the videos.

3.3 Failure analysis

While evaluation metrics can show us the how well a system is performing generally, we wanted to gain further insights into their strengths and weaknesses. For this reason a failure analysis of the best performing model was performed.

This analysis also serves our goal of making the work more explainable. As discussed in section 1.1.4, we chose less complex models for this work for their inherent explainability factor. Together with this failure analysis, this yielded a deeper understanding of why the model behaves like it does and how this can be improved upon.

4 Methods

4.1 Dataset

For this research, the Corpus NGT (CNGT) dataset has been used [20], [21]. This corpus contains videos of conversations between pairs of signers in NGT. CNGT contains a total of 2375 sessions, 103 of which have been annotated with head movements (head-shakes and head sways). In total, 2264 moments have been annotated with head movement, most of which are head-shakes. The average length of these annotations is 2.4 seconds (std 5.3), which means that most annotations are short, while some are tens of seconds long, with outliers of more than a minute.

For every session in the dataset, there exist two recordings: one for each of the signers. Every video is recorded at a resolution of 1280x720p at 25 frames per second. Every video begins and ends with a short display of the origin of the video with a creative commons license, these frames where discarded for the purpose of analysis.

During annotation the following definition of head-shakes and head sways was used: “Head-shakes and head sways share a movement of (part of) the face in the video image from left to right. Shakes involve rotation of the head about the top-bottom axis, sways involve movement of the head about the front-back axis. Sways of the head can also be the result of or co-occur with body sways; if the head also moves, then it should be annotated as Ns or Nsx, but not if only the body moves.” Figure 2 shows the distribution of the annotations. For this work all labels that involve head-shakes were aggregated, this means that no distinction was made between the different types of head-shake annotations for the purposes of training and evaluating the models.

During the annotation process, the annotators had the option to indicate doubt for any annotation that they were not sure of. Due to the majority of annotations being marked without doubt (see figure 2), the decision was made to exclude unsure annotations from the dataset. As a consequence, 4 more videos were not annotated with head-shakes, resulting in a total of 99 annotated videos.

A data split has been used to train and evaluate models during the course of the thesis. Due to the limited (annotated) data availability, a cross-validation approach was chosen to waste minimal data points during validation, at the cost of compute time. Specifically, 5-fold cross validation was used, creating splits at speaker level to ensure that the models did not overfit on the mannerisms of the speakers in the training footage. A hold-out test set has been used for a final evaluation after the conclusion of the development process. For this purpose, 10% of the head-shake annotations were set aside, again on the speaker level. This final evaluation was done by averaging the performance of the 5 models created during cross validation, this was done as to not pick an arbitrary fold.

4.2 Models

The following sections explain different techniques used to solve this problem in ascending order of complexity. All experiments were implemented using Python3.10 [22], all code is publicly available through GitHub [23].

4.2.1 Random baseline

For this technique, head-shake predictions were made at random to identify a lower bound for performance. This means that for any video, a random arrangement of background and head-shake predictions of the same length as the number of frames from the video was produced.

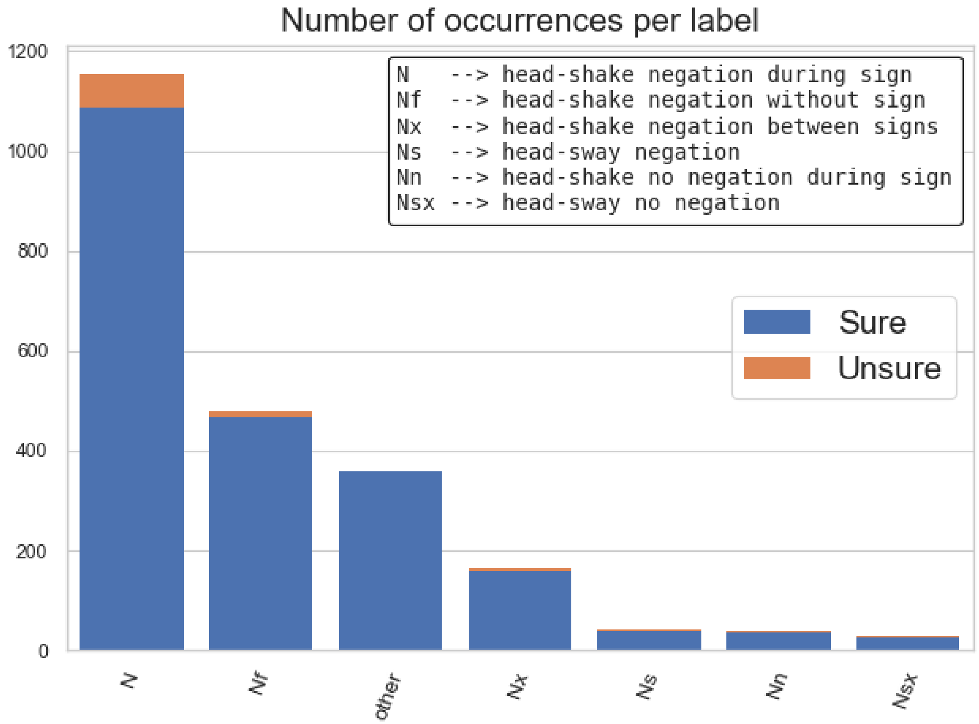

Figure 2: Number of occurrences for every head movement annotation in CNGT. The colours represent the level of confidence annotators indicated for the annotations. Annotations marked 'other' did not adhere to the annotation conventions.

4.2.2 Pose detection

Pose detections from two out-of-the-box pre-trained models were evaluated on CNGT:

1. OpenPose, a lightweight pose estimation system that used Part Affinity fields to track body parts in real-time. [24]
2. YOLO (You Only Look Once) is a popular detection system [25]. Later iterations of YOLO have released pre-trained pose estimation systems. For this research we have used the smallest pose estimation variant for the latest release (YOLOv8).

Because the pose detection systems are able to make an arbitrary number of predictions, the subject of the video was determined through the following strategy: for every frame, the confidence value of the target bounding box was averaged together with the mean confidence values of the facial keypoints (nose, left eye and right eye). The highest value was then selected as the signer of the video.

Pose detections were transformed into pitch, yaw and roll of the signers face. In the context of faces, yaw is used to describe rotation over the vertical axis, pitch describes movement over the horizontal axis up and down and roll describes movement over the horizontal axis from left to right. While a shake can be described using only the yaw of the face, including the other movements can tell the model a different type of movement than a pure head-shake is occurring, since yaw and roll movements should be minimal during a head-shake. For this transformation the following formulas were used, proposed by [26]:

$$\begin{aligned} roll &= \arctan\left(\frac{pt_{eyeL}.y - pt_{eyeR}.y}{pt_{eyeL}.x - pt_{eyeR}.x}\right) \\ yaw &= pt_{nose}.x\prime - pt_{nose}.x \\ pitch &= pt_{nose}.y\prime - pt_{nose}.y \end{aligned}$$

Where pt_{eyeL} and pt_{eyeR} refer to the x and y coordinates of the pose estimation of the left and right eye, while pt_{nose} refers to the x and y coordinates of the pose estimation of the nose. Primed coordinates refer to their location in the next frame. This results in a sequence of length $frames - 1$, to match the ground truth, the first frame annotation is discarded. Yaw and pitch depend on their pixel displacement to the next frame, while roll is calculated as an angle of the face that only depends on their current frame.

4.2.3 Memory-based classifier

To set a baseline for performance, a simple classifier was implemented that compares the distribution of facial movements. Instead of using pitch, yaw and roll to model facial movement, movement vectors were created by taking the maximum displacement in the four cardinal directions (up, down, left, right) of the nose and threshold it. This resulted in a vector of the same length as the video where every value could hold either of 5 states: the direction of maximum movement or a no movement value where the threshold was not exceeded. Preliminary testing showed the best results at a threshold value of 1 pixel. These vectors were adapted from the work of [19].

The memory-based classifier depended on training data to determine distributions, the following was performed under the cross validation scheme detailed above. All training vectors were separated using their class label, resulting in a collection of background and head-shake sequences. These sequences were concatenated and the distribution of movement values was taken by dividing the number of occurrences by the length of the concatenated sequence (total number of frames with that class label). This yielded two distribution vectors of length 5 for background sequences and head-shake sequences. During inference, a sliding window was used to predict the frame-level labels of the sequence by taking calculating the distribution of the movement values in the window and labeling it with the most similar training distribution using cosine similarity. Details of the sliding window approach are given in section 4.3.

4.2.4 HMM

For head-shake detection we implement a solution similar to [19]. We trained two HMMs: a head-shake model and a background model using the same number of hidden states as [19]: 3 for the head-shake model and 5 for the background model, the background model gets more states because it has to model more kinds of movements compared to the head-shake model. Instead of fitting the models using the movement vectors, we input a sequence of pitch, yaw and roll values calculated as described in section 4.2.2.

Consecutive frames annotated with the same label were extracted as input sequences for the models. Fitting and evaluation was performed using the cross validation regime described above. The models were fitted using the Baum-Welch algorithm, iterating over their respective datasets until convergence. During inference, a sliding window was used to predict the frame-level labels of the sequence by calculating the log probability of the window for both models. The model corresponding to the highest probability predicts the label for that position.

4.3 Sliding window

Both the memory-based baseline and the HMM approach make use of a sliding window to make predictions. This means that, rather than calculating a prediction over the entire sequence, a sequence of predictions was made over smaller sub sequences. This is necessary, because we want to use the log-likelihood of a sequence given a HMM, if we calculate this over the entire video we have no information on where the head-shakes are occurring. The sliding window approach requires two parameters: N for the size of the window and S for the stride (number of frames the window slides forward every step). The prediction over every sub sequence is mapped to the frame in the middle of the prediction window, this means that there are no predictions for the first and last $\frac{N-1}{2}$ frames. To avoid having to make arbitrary decisions as to which frame is considered the middle, the experiments were designed such that sliding windows of even N were increased to $N+1$. For this work we determined the optimal N empirically by repeating the experiment with different window sizes and measuring the effect on the F1-score. S was set to 1 to gain a prediction for every frame in the video fragment (apart from the start and end). Figure 3 shows an illustration of the process.

4.4 Evaluation metrics

To evaluate the performance of the models developed during this work, multiple evaluation metrics were chosen. These metrics are introduced and motivated in sections 1.1.7 and 3.2.

The evaluation metrics were calculated using the following definitions:

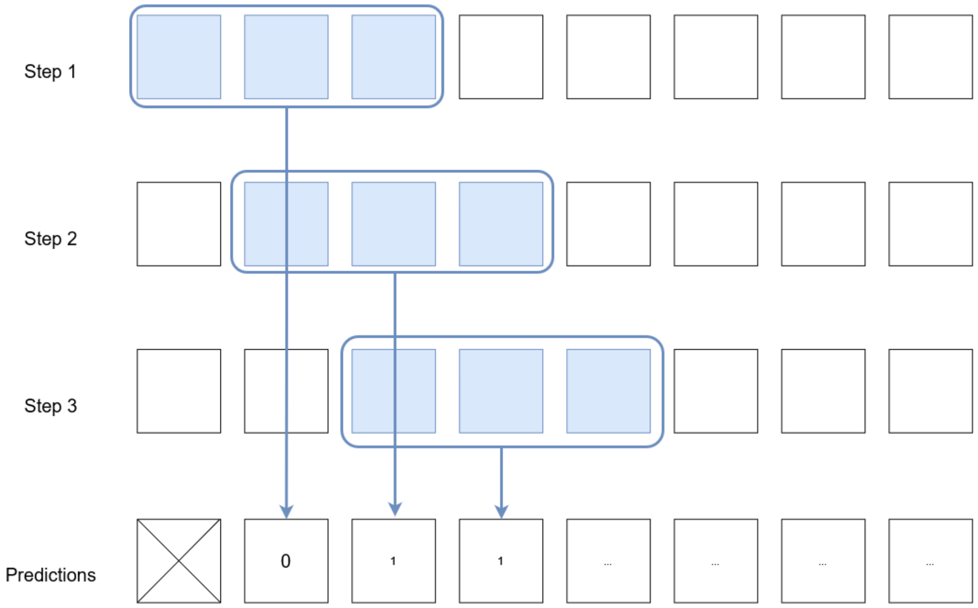

Figure 3: This figure shows an illustration of the sliding window approach. The boxes represent frames, the rows represent a video fragment. This example uses $N = 3$ and $S = 1$. For every step of the process, a single prediction is made and mapped to the middle of the sliding window. Notice that the first frame has no prediction.

$$\begin{aligned} Accuracy &= \frac{TN + TP}{TN + FP + TP + FN} \\ Precision &= \frac{TP}{TP + FP} \\ Recall &= \frac{TP}{TP + FN} \\ F_1 &= \frac{2}{Recall^{-1} + Precision^{-1}} \end{aligned}$$

The normalized Levenshtein distance (also known as the Levenshtein ratio) was used. For the calculations, equal weights were used for the three possible operations (additions, deletions and substitutions). This results in a number in the range 0-1 for every video fragment in the dataset.

Previously mentioned methods work well to evaluate the frame-level predictions of models developed during this research. As explained in section 3.2, for this work we were also interested in evaluating the problem by treating head-shakes as single events rather than collections of frames. Preliminary examinations of model output revealed that the HMM approach yielded clustered predictions rather than fragmented frames more often than the baseline methods. For this reason we considered the predictions suitable for an event-level analysis. Instead of considering frame level predictions, we consider all sub sequences in a video with the same prediction as a single event (both positive and negative).

Events were evaluated using a strategy adapted from [27]. This strategy introduces a tolerance window which essentially evaluates predicted events that start before the ground truth annotations as true positives. The intuition behind this approach is that the end user of the system (in our case linguists) need a few seconds to adjust to an event, so starting just before the annotation can be helpful to their understanding of the predicted event. We picked a tolerance window of 3 seconds, because users of the system need about 3 seconds to adjust to the viewing of a prediction [28]. The strategy evaluates events as follows:

1. True positive: a prediction is considered a true positive when their span falls in the window of a ground truth annotation, extended by a tolerance window.
2. False negative: a ground truth annotation is considered to be a false negative if no predictions fall within their window, extended by a tolerance window.
3. False positive: a prediction is considered a false positive when their span does not fall within the window of a ground truth annotation, extended by a tolerance window.
4. True negative: the space between ground truth annotations is considered a true negative when no predictions fall within their window, shortened by a tolerance window.

While the HMM method yielded more clustered predictions compared to the baseline methods, there were still small gaps in predicted events. To solve this problem, a smoothing filter was introduced. The filter moves over the prediction sequence with a sliding window and takes a majority vote for every prediction based on it's neighbours. The effectiveness of this filter depends on the size of the sliding window, which we empirically determined by calculating the event-level F1-score over the CNGT dataset.

To this end we propose a new performance metric: the flip ratio. We define this metric as the fraction of the number of predicted flips over the number of flips in the annotations of a video. A flip is a change of value in the prediction/annotation of consecutive frames in a video. This measures if the number of predictions (but not their location!) and their quality is comparable to the ground truth. If the predictions overlap perfectly with the ground truth, the flip ratio will be 1, lower values indicate that the systems missed predictions, while larger values either indicate an abundance of false positives or it can indicate that the predictions are fragmented and therefore result in more flips.

$$FlipRatio = \frac{n\_prediction\_flips}{n\_annotated\_flips}$$

4.5 Failure analysis

To gain insight into the strengths and weaknesses of the best model, as well as into the quality of the dataset and annotations, a systematic failure analysis was performed on the model with the highest F1-score. To this end, 40 videos were uniformly sampled with respect to the cross validation dataset. For every video, slowed-down fragments of all the true positive, false positive and false negative predictions were reviewed to identify patterns. The reviewer was allowed to replay the fragments as many times as they needed until they felt they understood what was going right and wrong for the predictions in any video. After reviewing the fragments, the entire video was reviewed once at normal speed, indicating true positive, false positive, true negative and false negative hits for every frame.

Model	Accuracy	Precision	Recall	F1 score
Random	0.50	0.05	0.50	0.09
Memory-based	0.66	0.10	**0.71**	0.17
HMM	**0.78**	**0.12**	0.54	**0.19**

Table 1: This table shows the performance of the systems developed to detect head-shakes in CNGT using different metrics. Evaluation was performed on the validation set using cross validation at frame-level.

5 Results

5.1 Pose detection

To create features for a head-shake detection system, pose estimation was performed on a the CNGT dataset. OpenPose was used to create pose estimations on the unobscured, annotated dataset. Visual inspection of the results revealed noticeable noise in the pose estimations, not remaining stable on the target areas even when the signer was sitting still. Due to the movements we focus on for this research, the decision was made to switch to a larger model. YOLOv8 nano was used to create pose estimations on the same dataset. Visual inspection revealed these estimations to be noticeably more stable.

5.2 Head-shake detection

Using cross-validation, 2 Hidden Markov models were trained to detect head-shakes in CNGT. To achieve this, a sliding window approach was used to determine the highest log likelihood for any given frame in the dataset, the details for this sliding window are described in section 4.3. To determine the optimal size for the sliding window used for collecting model predictions, a hyper-parameter search was performed by evaluating the F1-score over every frame prediction in the dataset. Figure 4 shows the results of this experiment. From the results it follows that 1.5 seconds is the optimal window size under these criteria and has been selected for all experiments described below.

5.3 Frame-level evaluation and sequence alignment

Figure 5 show the behaviour of different evaluation methods on CNGT on a video-level. The flip ratio shows that most videos have a noticeably higher number of prediction flips compared to the ground truth, with a few extreme outliers. We can see that performance modeled by the Levenshtein ration is most similar to calculating the accuracy. Precision varies significantly over the dataset while recall is generally low.

Table 1 shows the overall performance of the different methods on the frame level. We observe that both the memory-based and HMM approach outperform the random baseline. There is a trade-off between the memory-based approach and the HMM approach where the memory-based system achieves a higher recall at lower precision and vise-versa. Note that the performance of the memory-based system is further determined by the movement cut-off for the input sequence, which was optimized to achieve the highest F1-score. We use the F1-score to declare the HMM system slightly superior to the memory-based approach.

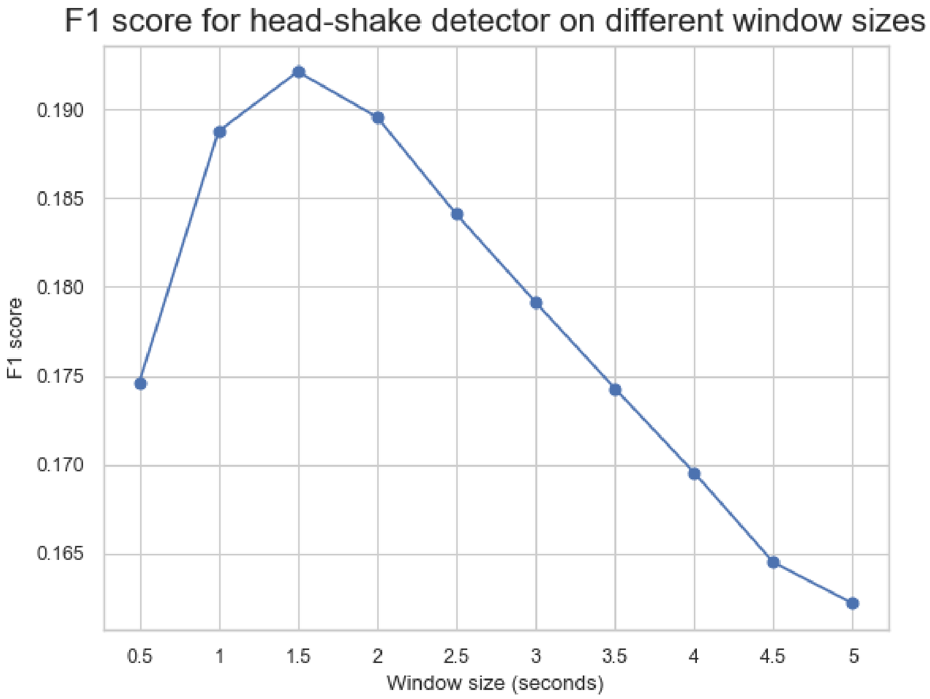

Figure 4: This figure shows the frame-level F1 score for head-shake detection on the CNGT dataset using different lengths for the sliding window. Evaluation was performed on the validation set using cross validation at frame-level.

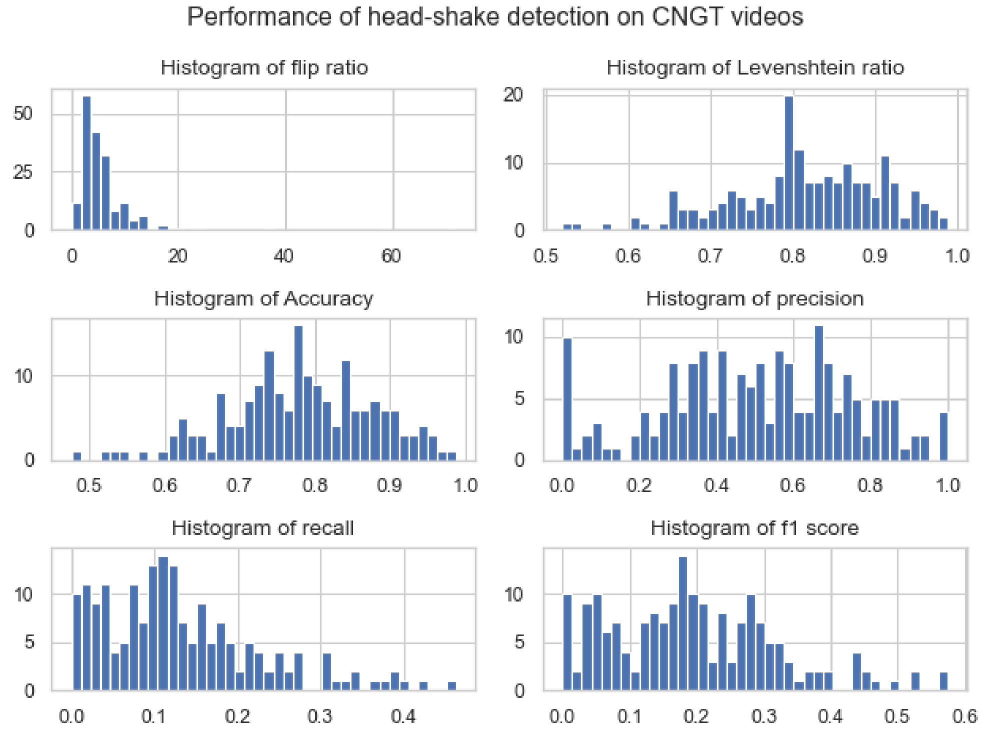

Figure 5: This figure shows histograms for various performance scores for head-shake detection in CNGT per video in the dataset. The scores have been calculated for every video individually on a frame-level. This means that the x-axis shows the performance on a number of videos indicated by the height the bins. Evaluation was performed on the validation set using cross validation at frame-level.

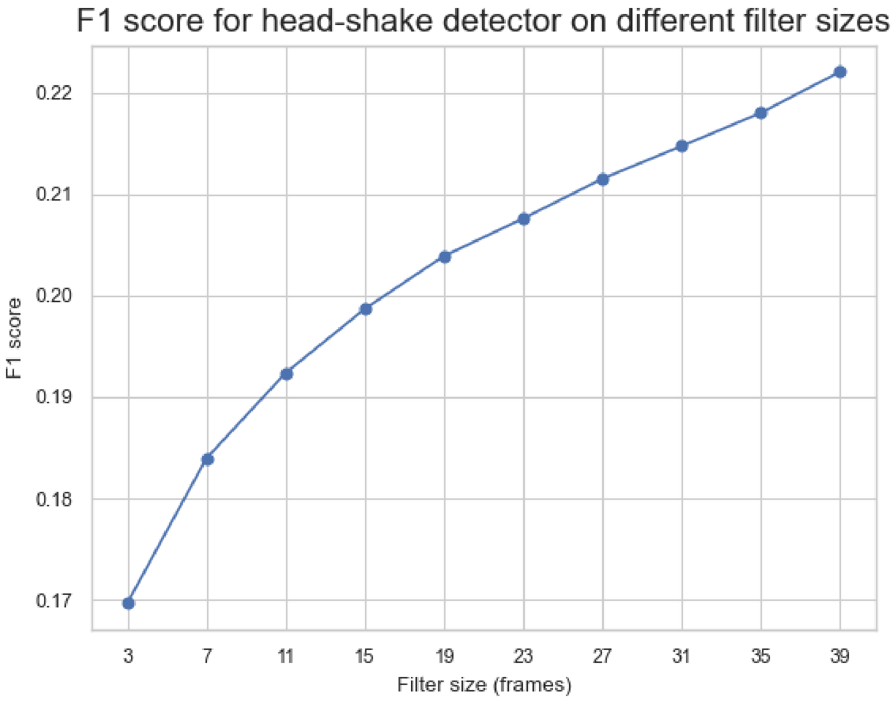

Figure 6: This figure shows the effect of the size of the smoothing filter on the F1-score when we evaluate the HMM head-shake detection approach on the CNGT dataset. Evaluation was performed on the validation set using cross validation at event-level.

Evaluation	Evaluation set	Accuracy	Precision	Recall	F1 score
Frame level	Validation	0.78	0.12	0.54	0.19
Frame level	Test	0.71	0.10	0.60	0.17
Event level	Validation	0.28	0.24	0.71	0.35
Event level	Test	0.26	0.21	0.77	0.36

Table 2: This table shows the performance the HMM based approach when evaluated on the frame level and the event level. The smoothing filter was applied to both evaluations. Evaluation was performed on the validation set using cross validation and on the test set by combining the predictions of the cross validation models.

5.4 Event-level evaluation

In an attempt to have the evaluation metric match the usefulness to linguists more, the system was evaluated at an event level. Section 3.2 and section 4.4 detail and motivate this evaluation system. Figure 6 shows the results of an experiment conducted to find the optimal filter size for the smoothing of the predictions. The results show that the F1-score of the events goes up beyond a reasonable filter size. To find a suitable filter size, bar codes of the predictions were evaluated. A filter size of 7 frames yielded the desired effect of closing fragmented gaps in events without connecting it to unrelated events or shifting the label position (this started occurring after a size of 15 frames) Table 2 shows the results of both evaluation methods. We observe that the reported quality of the systems change drastically depending on how we interpret the results. For event-based evaluation, the system is more precise and yields a higher recall, while frame based evaluation shows a noticeably higher accuracy. For the test set, precision is slightly lower and recall is slightly higher for both evaluation methods.

5.5 Speaker sensitivity

To evaluate the sensitivity of the system on a speaker level, we aggregated the results of the frame-level evaluation per speaker using different metrics and performed a one-way ANOVA test on these distributions to determine if there is statistically significant sensitivity to the signer for head-shake prediction. The results are shown in figure 7 (F1), figure 8 (recall) and figure 9 (precision). We observe p-values of 0.021, 0.001 and 0.007, respectively. Which show us that the performance of the system is significantly different depending on the speaker in CNGT under the assumption that p-values below 0.05 can be considered significant.

5.6 Failure analysis

To understand the behaviour of the head-shake detection system and promote explainability of the work, a failure analysis was performed on a uniformly sampled subset of NGT conversation videos. Section 4.5 detail the process by which the analysis was performed. This section describes the observations made during this process.

5.6.1 Movement sensitivity

The most obvious observation during the failure analysis was that the vast majority of the model's predictions involved movement. Either because the person was communicating signs to the other or because they expressed understanding while observing the signs of the other.

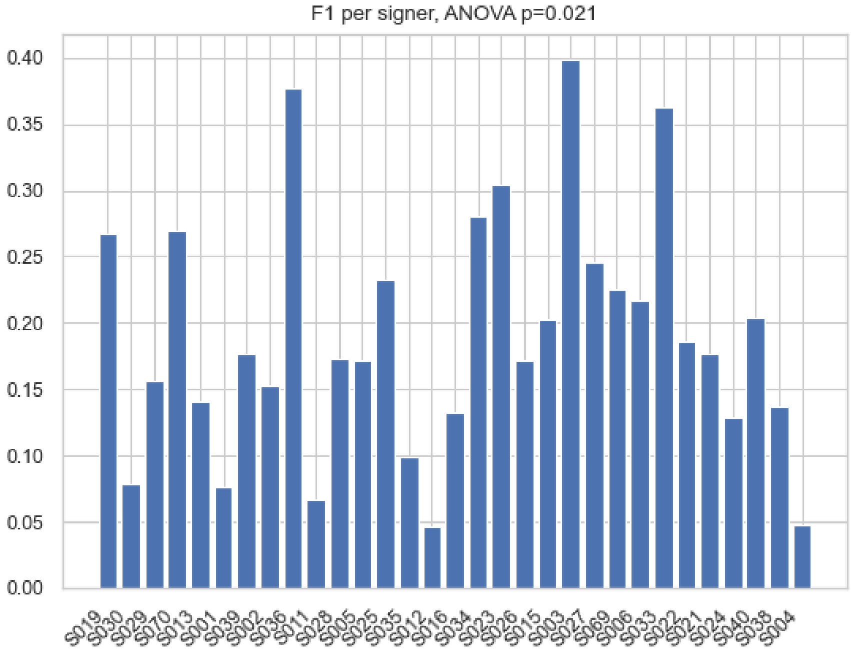

Figure 7: This figure shows the F1 score per signer in the CNGT dataset evaluated on frame-level using cross validation on the validation set.

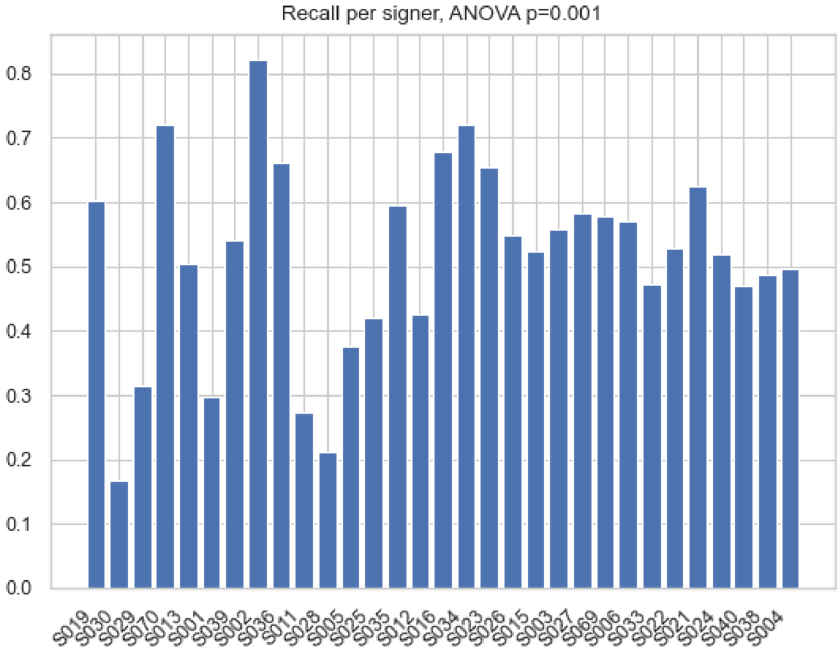

Figure 8: This figure shows the recall per signer in the CNGT dataset evaluated on frame-level using cross validation on the validation set.

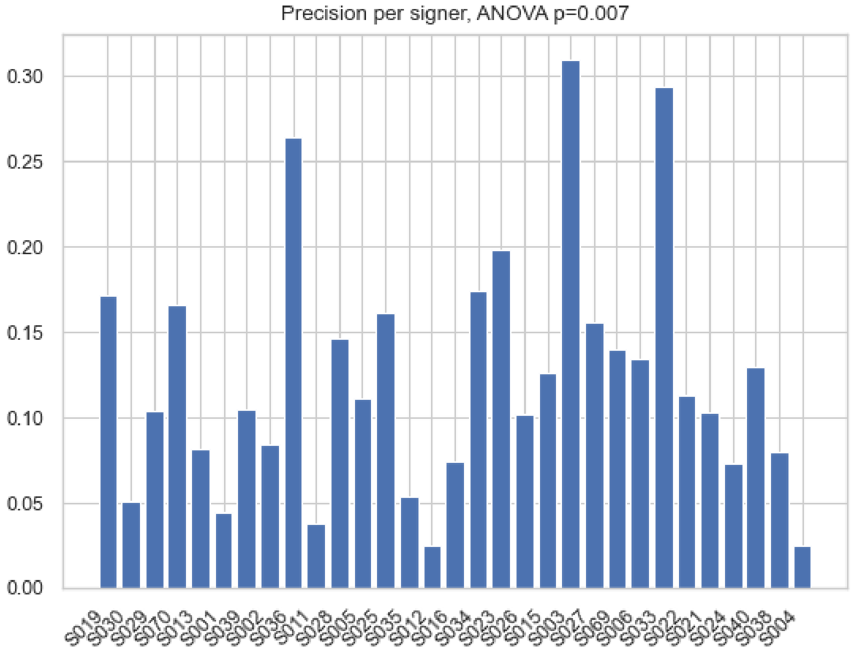

Figure 9: This figure shows the precision per signer in the CNGT dataset evaluated on frame-level using cross validation on the validation set.

5.6.2 Left-to-right movements

The system triggered on left-to-right head movements that did not constitute a head-shake. These movements happened for several reasons. For example, a signer might turn their head sideways as part of the communication. There were also instances where an intervention was made by the researchers where the participants would turn their heads to ask or answer a question.

By consulting a domain expert, we learned that some of instances where left-to-right head movements are made are part of the sign language itself. Signers can look into different directions to communicate that they are discussing different locations, people or events. Even a short eye gaze can be enough to communicate this in sign language.

5.6.3 False positives

There were many instances where the signer would be conveying a movement that involves the head which did not resemble a head-shake, but still yielded a positive prediction from the system.

5.6.4 Sensitivity to explicit head-shakes

The system was better able to identify extreme head-shakes. With this we describe head-shakes where the face moves further to the sides or repeats the shake movement more often. Subtle head-shakes were more often missed by the system.

5.6.5 Head-nod triggers

Interestingly, the system had a tendency to trigger on head-nod movements (where the head repeats movements between up and down). While labels did exist in the dataset for head-nod movements, these were not used during training.

5.6.6 Missing annotations

Surprisingly, during failure analysis, false positive predictions were identified that appeared to convey a legitimate head-shake movement. Because of this observation, another experiment was conducted where some of these predictions were sampled and sent to a domain expert for confirmation, we describe the results of this experiment in section 5.7.

5.6.7 Rapid head-shakes

Finally, it was observed that the system had a hard time detecting head-shakes that occurred very rapidly. In contrast, head-shakes that were performed very slowly and deliberately by the signer were often picked up by the system.

5.7 Expert opinion

To corroborate the observation that the system picks up on annotations missed by the annotators of the dataset, 30 videos were uniformly sampled from the dataset and checked for potential missed annotations. Through this process, 10 potential cases were identified and sent to an expert to determine if these fragments were linguistically interesting. The expert reported 2 of these cases as false positives and 8 as head-shakes, 6 of which where involved with negation, the other 2 were involved with conveying an emotion. The conclusion was made that such movements should be picked up by a good detection system, then an annotator can manually decide for every case if it is relevant to their research goals or not.

6 Discussion

The results of this work indicate that the detection system does not work flawlessly, it makes many false positive predictions and misses a number of annotations from the dataset. It is clear that head-shake detection in NGT conversation footage is not a trivial task. One explanation of this is that only one of the Hidden Markov models is trained on specific movements: the head-shake model, while the background model is trained on every other movement which occurs in the hours of video footage, this makes for a very general model which is not able to discriminate well in a setting where the label distribution is as unbalanced as CNGT.

Because the data other papers [17, 19] used for implementing similar solutions to head-shake detection are not publicly available, we can only speculate as to why our detection results are noticeably lower. Since these papers created their own in-house datasets by performing shake and nod movements in front of a camera, it is possible that the examples in their datasets were more explicit. As we have learned from this work, explicit movements are more easily detectable than subtle head-shakes. In addition, these works evaluated their systems with a single prediction per example video fragment, instead of implementing a way to create frame-level predictions over a continuous video.

We have learned that the way we evaluate this task can significantly alter the perception of the resulting performance. This highlights the importance of studying complex tasks from multiple angles, rather than reporting a single performance metric. For this task, we argue that evaluating at the event-level gives a performance indication that is better aligned with the needs of linguists. However, we have also learned that this method of evaluation comes with limitations, namely that we can increase the scores on the dataset simply by creating unrealistically large predictions with the smoothing filter. Adjustments to the evaluation method could be an interesting direction for future research. For example, the criteria for determining a true positive could be made more strict by restricting the amount of frames a prediction is allowed to exceed the boundaries of the ground truth annotation. Alternatively, one could cut the negative annotations into smaller fragments, since entire minutes of footage where no head-shake occurs are now condensed to a single event.

We have observed significant variance in performance depending on the signer. One explanation for this could be the difference in experimental set-up between conversations. Lighting conditions, camera angle and distance and signer orientation vary from signer to signer in CNGT. It should also be noted that there are significantly more video of certain signers in the dataset, while others only appear in a handful of videos, which means that a difference in performance for some signers could be explained by small sample size. A possible strategy for future work could be to cluster the dataset or states of the predictive models in such a way that if shows variance in the performed head-shakes, while staying invariant to the signer. Alternatively, one can take inspiration from other fields such as sign language recognition that have an existing body of research into decorrelating feature representations from speakers using speaker labels (which are available for CNGT) using adversarial training [29], [30].

Several interesting observations were made while evaluating the predictions of our head-shake detection system. Importantly, the system was triggered by left-to-right movements. For instance because a singer is laying emphasis that they are describing separate locations, which is linguistically interesting. But also because the signer is responding to or asking a question to the researchers filming the footage. Additionally, we have learned that the system is generally sensitive to any kind of head movement, rather than shakes specifically. It is also better at detecting explicit head-shakes as compared to subtle ones. Analysis of the predictions also revealed that head-shakes exist in the dataset which have been missed by the annotators.

Through a practical example, we have shown that even though the system is not optimal, it can already be used to assist linguists in their research. Through observing just the predictions of our system, we were able to identify head-shake cases missed by the annotators of the dataset. This shows how such systems can be used to refine dataset annotations with minimal time investment (analysing short prediction fragments rather than hours of footage). If future work is able to refine the system further, it could even be used to speed up the annotation of entirely new videos, and most of CNGT has not been annotated for head movement at the time of writing this work.

The results for the validation and test set were slightly different. This is to be expected, in total the test set contained 14 videos over a dataset containing 99 videos. The difference is not large enough to warrant a concern that the models trained during this research were over-fitted on the validation dataset.

6.1 Limitations

As with all research, there are limitations to this work which have to be considered. One important limitation is that this work researched the role of non-manuals in NGT, signer behaviour across languages can vary significantly. For future work it would be interesting to see these results reproduced in an international context, which would require more datasets to be annotated.

Another limitation of the work is the evaluation of the pose estimation. No pose annotations exist for the CNGT dataset, so there is no objective numeric measurement of quality for the pose detection methods used in the research. Future work could improve this by annotating and evaluation the dataset with keypoints. In addition, it could be interesting to measure the impact of different pose detection systems on the performance of the head-shake detection system.

An interesting approach for future research could be to move on from foundation models and train systems on the video footage directly. An example of such a system could be [31], which was designed for sign language recognition by modeling the input videos as sequential feature vectors using a three-dimensional CNN and then forwarding this sequence to an LSTM based encoder-decoder, which is then aligned with the output of a CTC decoder. Since video data occurs over time, it makes conceptual sense to make a sequential prediction of whether or not a head-shake is occurring at every time point. A benefit of this approach is that it can be easily extended to predict other non-manuals in a single end-to-end system.

7 Conclusion

7.1 Research question 1

Can we automatically detect head-shakes from NGT conversation video footage using by leveraging pre-trained models?

While it is possible to do head-shake detections on NGT video conversation footage, we have shown that doing so is no trivial task, because simple methods and statistical models are insufficient to reach performance level comparable to expert annotators. This work set out the first steps towards accurate detection, but improvements are needed before the system can be considered reliable.

7.2 Research question 2

Do head-shake detection models trained on NGT data focus on the same features a human would focus on?

From evaluating the predictions of our system we have identified that there is a disproportionate focus on movements unrelated to head-shakes. Because we deliberately chose to use statistical models that use relevant features as input, we know the predictions the models made are not based on features unrelated entirely to head movement.

7.3 Research question 3

Are head-shake detection models sensitive to signers in NGT?

We have found statistical evidence indicating that the predictions of our system are sensitive to the signer. We have suggested various reasons to explain this, most importantly dataset imbalance and varying conditions in the creation of the conversation footage.

7.4 Research question 4

How can head-shake detection systems be used to facilitate linguistic research?

Through a practical example we have shown how non-manual detection systems can help linguists clean up dataset annotations and make discoveries of what can be considered to be a linguistically interesting non-manual.

References

[1] P.-A. Peia, L. Marty, T. Corrie, and S. Jan, “Literatuurstudie naar de Leefsituatie van Vroegdove Volwassenen,” *Koningklijke Kentalis*, 2016.

[2] R. Cokart, S. Trude, C. Tijsseling, and E. Westerhoff, *In pursuit of legal recognition of the sign language of the Netherlands. Chapter from The legal recognition of sign languages: advocacy and outcomes around the world.* Multilingual Matters, 6 2019.

[3] “StatLine - Bevolkingsontwikkeling; maand en jaar,” 2023.

[4] “Wet erkenning Nederlandse Gebarentaal (BWBR0045012),” 7 2021.

[5] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, “Explainable AI: A Review of Machine Learning Interpretability Methods,” *Entropy 2021, Vol. 23, Page 18*, vol. 23, p. 18, 12 2020.

[6] R. Pfau, J. Quer, *et al.*, *Nonmanuals: their grammatical and prosodic roles.* na, 2010.

[7] M. Oomen and R. Pfau, “Signing not (or not): A typological perspective on standard negation in Sign Language of the Netherlands,” *Linguistic Typology*, vol. 21, pp. 1–51, 7 2017.

[8] A. Chizhikova and V. Kimmelman, “Phonetics of Negative Headshake in Russian Sign Language: A Small-Scale Corpus Study,” *European Language Resources Association (ELRA)*, pp. 20–25, 2022.

[9] T. Johnston, “A corpus-based study of the role of headshaking in negation in Auslan (Australian Sign Language): Implications for signed language typology,” *Linguistic Typology*, vol. 22, pp. 185–231, 8 2018.

[10] S. C. Agrawal, A. S. Jalal, and R. K. Tripathi, “A survey on manual and non-manual sign language recognition for isolated and continuous sign,” *International Journal of Applied Pattern Recognition*, vol. 3, p. 134, 9 2016.

[11] O. Aran, I. Ari, L. Akarun, B. Sankur, A. Benoit, A. Caplier, P. Campr, A. H. Carrillo, and F. X. Fanard, “SignTutor: An interactive system for sign language tutoring,” *IEEE Multimedia*, vol. 16, pp. 81–92, 1 2009.

[12] H. Brock, I. Farag, and K. Nakadai, “Recognition of non-manual content in continuous japanese sign language,” *Sensors*, vol. 20, no. 19, p. 5621, 2020.

[13] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, “2D Human Pose Estimation: New Benchmark and State of the Art Analysis,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[14] S. R. Eddy, “What is a hidden Markov model?,” *Nature biotechnology*, vol. 22, no. 10, pp. 1315–1316, 2004.

[15] L. Rabiner and B. Juang, “An introduction to hidden Markov models,” *ieee assp magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[16] T. Starner, *Visual recognition of american sign language using hidden markov models.* PhD thesis, Massachusetts Institute of Technology, 1995.

[17] Y. G. Kang, H. J. Joo, and P. K. Rhee, “Real time head nod and shake detection using HMMs,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4253 LNAI - III, pp. 707–714, 2006.

[18] R. Caruana and A. Niculescu-Mizil, “Data mining in metric space,” pp. 69–78, Association for Computing Machinery (ACM), 8 2004.

[19] H. Wei, P. Scanlon, Y. Li, D. S. Monaghan, and N. E. O’Connor, “Real-time head nod and shake detection for continuous human affect recognition,” *International Workshop on Image Analysis for Multimedia Interactive Services*, 2013.

[20] O. Crasborn, I. Zwitserlood, and J. Ros, “The Corpus NGT. An open access digital corpus of movies with annotations of Sign Language of the Netherlands,” 2008.

[21] O. Crasborn, T. Hanke, E. Efthimiou, E. Zwitserlood, and E. Thoutenhoofd, “The Corpus NGT: an online corpus for professionals and laymen,” pp. 44–49, 2008.

[22] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual.* Scotts Valley, CA: CreateSpace, 2009.

[23] “C. Rijbroek, Sign-Language-Thesis (software), accessed through https://github.com/Casvanrijbroek/Sign-Language-Thesis.”

[24] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, “Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields,” in *CVPR*, 2017.

[25] G. Jocher, A. Chaurasia, and J. Qiu, “YOLO by Ultralytics,” Jan. 2023.

[26] E. N. A. Neto, R. M. Duarte, R. M. Barreto, J. P. Magalhães, C. C. Bastos, T. I. Ren, and G. D. Cavalcanti, “Enhanced real-time head pose estimation system for mobile device,” *Integrated Computer-Aided Engineering*, vol. 21, pp. 281–293, 2014.

[27] N. Hollain, M. Larson, and F. Roelofsen, “Distractor-Based Evaluation of Sign Spotting,” *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pp. 1–5, 6 2023.

[28] A. P. De Vries, G. Kazai, and M. Lalmas, “Tolerance to irrelevance: A user-effort oriented evaluation of retrieval systems without predefined retrieval unit,” in *RIAO 2004 conference proceedings*, pp. 463–473, 2004.

[29] C. Yang, S. Wang, X. Zhang, and Y. Zhu, “Speaker-Independent Lipreading With Limited Data,” pp. 2181–2185, IEEE, 10 2020.

[30] H. Li, M. Tu, J. Huang, S. Narayanan, and P. Georgiou, “Speaker-invariant affective representation learning via adversarial training,” vol. 2020-May, pp. 7144–7148, Institute of Electrical and Electronics Engineers Inc., 5 2020.

[31] J. Pu, W. Zhou, and H. Li, “Iterative Alignment Network for Continuous Sign Language Recognition,” pp. 4165–4174, 2019.