Lexical resources for sign language synthesis

Lyke D. Esselink

Layout: typeset by the author using LaTeX.
Cover illustration: Final product spelling the word ‘Thesis’

Lexical resources for sign language synthesis

The translation of Dutch to Sign Language of the Netherlands

Lyke D. Esselink
11175281

Bachelor thesis
Credits: 18 EC

Bachelor *Kunstmatige Intelligentie*

University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

Supervisor
Dr. F. Roelofsen

Institute for Logic, Language and Computation
Faculty of Science
University of Amsterdam
Science Park 907
1098 XG Amsterdam

June 26th, 2020

Acknowledgements

I would like to express my appreciation to my supervisor Dr. Floris Roelofsen, for all the time and effort that went into supervising this project, and for going above and beyond to help find resources and improve the overall project.
I would like to thank Dr. Inge Zwitserlood, Dr. Onno Crasborn, Johan Ros, and the rest of the staff and organisations of the VisiCast, eSign, Gebarennet, LinguaSign, and Automated Sign Language (Prins and Janssen, 2014) projects. Due to the work of these people, we were able to work with a much larger database of signs than we originally hoped, as they provided us with the material to make this possible. Additionally, I would like to thank Dr. Sarah Ebling for providing help with SiGML notation of some signs that were not in the lexicon. Furthermore, I would like to thank Dr. John Glauert and Dr. Richard Kennaway for their help with the JASigning software and SiGML notation. Lastly, thanks goes out to Mw. Marijke Scheffener, who has been an enormous help with teaching me NGT this past year and the evaluation of the results of this project.

Abstract

Deaf people oftentimes have no choice but to communicate in the *lingua franca* of their country, despite the fact that this is a second language for them. Their ability to read and write is significantly worse than that of a hearing person. Additionally, if a deaf child is born into a family of hearing parents, it is vital that the parents communicate with the child through sign language as soon as possible, otherwise the language region of that child's brain will be underdeveloped. There are currently few resources available for people to learn Sign Language of the Netherlands (NGT). A tool that makes this possible would be invaluable to those who need it.

This thesis investigates what components are necessary to build a system that uses an avatar to translate a Dutch sentence to NGT. On the basis of the workings of NGT and previous research on the matter, three main components are devised and implemented: software for sign language synthesis, lexical resources for encoding signs, and grammatical resources for structuring sentences. Results show a proof of concept with a corpus of 2226 signs, the ability to make correct use of classifiers, and the abilities to fingerspell any word that is not in the corpus and convey numbers correctly. However, the corpus contains a number of errors and of which manual errors and unnatural mouth signs should particularly be improved. Moreover, further research is necessary to improve the comprehensibility of signs.

Preface

This paper describes an individual contribution to a larger project executed in close collaboration with Corsel (2020) and Mende-Gillings (2020). The first parts of these three papers describe the overall project and were jointly written, making them largely identical. Section 1.2 describes the global research question, and chapters 2 and 3 provide theoretical context and set up a hypothesis. Section 5.1.1 shows the overall program created, including the components of the other projects, and section 5.2.1 evaluates the result. These previously mentioned chapters and sections have been written in joint collaboration with Corsel (2020) and Mende-Gillings (2020).

An introduction to and motivation for the overall project are provided in section 1.1. Chapter 4 discusses and addresses the goals specific to the research of this paper only, and sections 5.1.2 and 5.2.2 pertain to the results and the evaluation of this research. Finally, in chapter 6 a conclusion and discussion of this research is given. These chapters and sections regard individual efforts, and have therefore been written individually. For more specific information about the other components of this project, the reader is advised to read the other two aforementioned papers. However, it is not necessary to have read them to understand this paper.

Contents

Chapter 1

Introduction

1.1 Context

Whether it is written, spoken, signed, or otherwise, language and communication play an important role in the life of every living person. Although writing and reading is taught at a young age, it is not a simple matter. Reading comprehension consists of two separate elements: *linguistic comprehension* and *decoding*, the former being "the ability to take lexical information (i.e., semantic information at the word level) and derive sentence and discourse interpretations", and the latter comprising "efficient word recognition ... the retrieval of semantic information at the word level" (Hoover and Gough, 1990). Decoding is "closely related to phonological skill while linguistic comprehension depends upon general language skill", and both are necessary for reading (Wauters et al., 2006). A majority of Deaf[1] people therefore struggle with their reading and writing abilities, as they lack the phonological skill required to do so (Wauters et al., 2006). The level of reading comprehension of a Deaf student aged 17–18 lies around the same level of that of a hearing student aged 9–10 (Holt, 1993). Deaf people rely on sign language as a better form of communication, which should be seen as their first language, whereas the written language of their country should be considered a second language (Kennaway et al., 2007).

For some, the need to learn sign language can arise naturally: people that are born deaf might be exposed to it their entire life. Others' hearing abilities may diminish slowly, giving them the time to gradually immerse themselves in a language. However, this is not the case for all people: a deaf child might be born into a hearing family, resulting in a great and immediate need for the family members to learn sign language. This is especially important as "these early interactions establish the foundation upon which language develops", and "the fluency level that [children that are taught sign language at around five years or older] reach in their sign language may be considerably lower than that of children who have been offered sign language from birth, but also lower than that of hearing people who have learned a sign language at an even later age" (Marschark, 2001; Baker et al., 2016).

Considering the fact that sign languages are all so intricate, there are certain barriers when attempting to learn one. One of which is that sign languages tend to be be very inaccessible. For Sign Language of the Netherlands (NGT), the only resources available are two extremes: on one side there are extremely short classes which do not provide enough information to even hold a conversation, while on the other side there are courses that educate the student to become fully fledged translators. There are almost no alternatives, and there is no single source available that can quickly provide a correct translation of a sentence, even though this would prove useful.

One important fact to consider is that the sign language of a country and the spoken/written language of a country are two completely distinctive languages. Each has its own collection of words and grammatical rules. Deaf people often have to read or write in their second language, which can prove to be a challenge. As a result, they often rely on a interpreter for more efficient communication with people that are not familiar with sign language. Interpreters can be scarce and may not be available for every situation in which they might be useful or necessary. In the last decades, there has been an increasing interest in animated sign language avatars, which can serve as translators in lieu of human interpreters

[1] The upper case word 'Deaf' is conventionally used to describe members of the linguistic community of sign language users, while the lower case word 'deaf' refers to the audiological state of being deaf (Morgan and Woll, 2002)

when these are unavailable.

1.2 Aim

An animated sign language avatar would be an attainable solution for this lack of resources, the need for an accessible way for people to learn sign language, and the scarcity of human interpreters. The goal of the project is to build the basis for a tool in which a user can enter a Dutch sentence and in return see a correct translation of this sentence in NGT by an animated avatar. This will prove a valuable resource for people attempting to learn a sign language and/or communicate with a Deaf person. The project aims to answer the question "What are the necessary components for a system that uses an avatar to translate a Dutch sentence to Sign Language of the Netherlands?".

Part of the question, namely how to translate from Dutch to NGT, has been covered in previous research (Weille, 2019; Brinkhuijsen, 2019; Smit, 2019). Therefore, this research will focus on the production of NGT from glosses, which are textual representations of signs (e.g. HOUSE[2] is the gloss of the sign that means 'house'). The goal of this project is to create a proof of concept with a limited vocabulary that allows a user to input a glossed NGT sentence, whose translation will be signed by an avatar. This avatar would display all the attainable essential elements of the translation.

The remainder of this paper is structured as follows: chapter 2 discusses the theoretical foundation of the workings of NGT and previous research done in the field of sign language translation and synthesis. In chapter 3, a hypothesis for the research question is formed on the basis of chapter 2, and a global methodology is described for the development of the necessary components. Chapter 4 focuses on the lexical resources for sign language synthesis by formulating specific goals for its development, and illustrates the implementation of these goals in detail. In chapter 5, both global and individual results are presented, analysed, and evaluated. Finally, chapter 6 delivers a summary and critical reflection on the approaches and results, and discusses open issues and future work.

[2]Glosses are indicated by capital letters to illustrate that these words are signed and not spoken

Chapter 2

Theoretical Foundation

It is a common misconception that there is a universal sign language. Although similarities can be found in signs for words that are globally characterised in the same manner (e.g. the sign HOUSE is identical in multiple sign languages — see Figure 2.1), parallel to spoken languages, sign languages have evolved independently of each other and impose their own distinctive set of grammatical rules (Emmorey, 2001). This is clearly illustrated by American Sign Language and British Sign Language: “despite the fact that [they] are surrounded by the same spoken language, they are mutually unintelligible”, and therefore “most often named for the country or area in which they are used”, as they are so distinctively unique (Emmorey, 2001). Sign Language of the Netherlands (NGT) is no exception to this rule: it even has five official dialects (Schermer and Koolhoof, 1991).

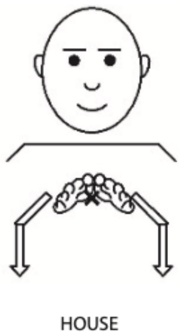


Figure 2.1: The sign HOUSE (Baker et al., 2016)

2.1 History of Sign Language of the Netherlands

Five Deaf institutes have been established in the Netherlands between 1790 and 1911, in different regions of the country. However, the use of sign language in Dutch Deaf-education was banned between 1915 and 1980 (Schermer and Koolhoof, 1991). This ban was due to the concept of ‘oralism’, “the practice of teaching Deaf students through spoken language using amplification devices and lip-reading, to the exclusion of all sign language communication” (Morrissey and Way, 2005). It was believed that it was better for deaf children to be educated in solely spoken language, as the usage of signs would inhibit the development of spoken language (Schermer and Koolhoof, 1991). Nevertheless, Deaf people still signed amongst each other. The combination of the previous and the fact that the five Deaf institutes in the Netherlands barely communicated with each other at the beginning of the 20th century caused the different dialects in NGT (Schermer and Koolhoof, 1991). Currently, NGT is used in Deaf-education and recognised as a language both politically and socially, but not legally (Schermer and Koolhoof, 1991; de Groot, 2019). The Deaf community is taking action to legally recognise NGT as an official language in the Netherlands (de Groot, 2019).

Figure 2.2: The signing space (Baker et al., 2016)

2.2 Grammar of Sign Language of the Netherlands

NGT, like all other sign languages, is a visual-spatial language: it is “articulated by using the hands, face, and other parts of the body, and all these articulators are visible ... signs are articulated on the body or in space close to the body” (Baker et al., 2016). Figure 2.2 shows the signing space used in NGT, which will be explained in section 2.2.4.

2.2.1 Phonology

The phonology of NGT consists of four aspects: handshape, orientation, location, and movement. When discussing sign language, the term *phonology* is used “despite its sound-based etymology, in order to emphasise that the same level of structure exists in spoken language” (Morgan and Woll, 2002).

(a) A ‘c’ handshape

(b) A ‘v’ handshape

(c) A ‘t’ handshape

Figure 2.3: Different kinds of handshapes (Baker et al., 2016)

Handshape

For handshape, there is a distinction between which fingers are selected (active) and the position of these fingers (Baker et al., 2016). Examples are shown in Figure 2.3. In Figure 2.3a, all four fingers are selected; in Figure 2.3b, the index- and middle finger are selected; and in Figure 2.3c, the little finger, ring finger, and middle finger are not selected, as they are not the ‘active’ fingers in this case. Selected fingers “can make contact with the body, the head, or the other hand and arm; can adopt a special position (curved, bent, closed, spread); can move (open and close)” (Baker et al., 2016). The position of selected fingers describe: “curving of the fingers [Figure 2.3a]; spreading of the fingers [Figure 2.3b]; an aperture relation between the thumb and the selected fingers [Figure 2.3c]” (Baker et al., 2016).

(a) The orientation for the sign EASY

(b) The orientation for the sign SUPPOSE-THAT

Figure 2.4: Different kinds of orientations in NGT (Baker et al., 2016)

Orientation

The orientation of a sign can be described by "identifying the part of the hand that points towards the location of the sign", the parts of the hand being the "palm, the back of the hand, the thumb side, the little finger side, the wrist side, and the tips of the fingers" (Baker et al., 2016). In Figure 2.4, the location of the two different signs is the same (the chin), but in Figure 2.4a, the palm points to the location, and in Figure 2.4b, the thumb side of the hand points to the location (Baker et al., 2016).

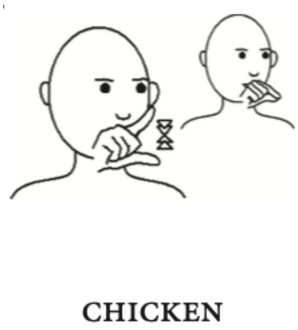
(a) The location for the sign CRUEL

(b) The location for the sign SWEET

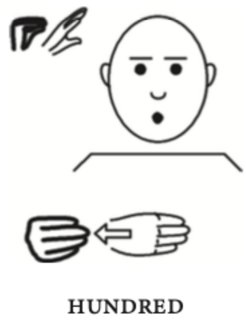
Figure 2.5: How location can affect the meaning of a sign — British Sign Language (Baker et al., 2016)

Location

The location of the sign is where in the signing space (Figure 2.2) the sign is articulated. There are four main locations: "the head, the upper body, the non-dominant (or weak) hand, and the neutral space", the neutral space being the space in front of the body (Baker et al., 2016). The location of a sign is an important distinction for its meaning. An example can be seen in Figure 2.5, where Figure 2.5a displays the sign for CRUEL, and Figure 2.5b the sign for SWEET: two contrasting meanings, yet almost identical signs.



(a) The movement for the sign CHICKEN



(b) The movement for the sign HUNDRED

Figure 2.6: Different kinds of movements in NGT (Baker et al., 2016)

Movement

Movements can be divided into two types: "movements of the fingers and wrist (hand-internal movements and orientation changes) and movements of the entire hand (path movements)" (Baker et al., 2016). One example of a path movement is the sign SUPPOSE-THAT in Figure 2.4b, where the whole hand moves towards the chin. Figure 2.6 shows two additional examples: Figure 2.6a illustrates hand-internal movements (only the fingers move), and Figure 2.6b displays a combination of a path- and a hand-internal movement: the fingers close while the entire hand moves (Baker et al., 2016).

(a) The sign BROTHER and SISTER (Verlinden and Zwitserlood, 2016)

(b) Non-manuals for the sign IDIOT (Baker et al., 2016)

(c) Non-manuals for the sign WANT-NOT (Gebarencentrum, 2020)

(d) Non-manuals for the sign WHY (Verlinden and Zwitserlood, 2016)

Figure 2.7: Non-manuals in NGT

2.2.2 Non-manuals

Non-manuals are composed of "form elements that relate to the posture of the body and the head, facial expressions, and certain movements or configurations of the mouth", and are vital components of a sign (Baker et al., 2016). Figure 2.7 illustrates four different types of non-manuals: mouthings, mouth gestures, a headshake, and general facial expression. Figure 2.7a shows the identical manual component for the signs BROTHER and SISTER, the distinction lies in whether the signer uses the mouthing 'broer' (brother) or 'zus' (sister). Figure 2.7b displays the mouth gesture accompanying the sign IDIOT: "a lax tongue hanging slightly out of the mouth while some air is being blown out" (Baker et al., 2016). Figure

2.7c accentuates the head shake for the sign WANT-NOT[1]: without it the sign would not be valid. Lastly, Figure 2.7d illustrates the slightly raised chin and furrowed eyebrows that are tied to asking a content question (WH-question) in NGT.

When asking a question in certain spoken languages, including Dutch and English, speakers use intonation to differentiate a question from a statement. A sentence like: 'I do the dishes', can be a statement when pitch is gradually lowered, but is a question if the pitch is gradually raised. NGT is similar, except instead of adjusting intonation, signers adjust non-manuals (Baker et al., 2016). To indicate that a sentence is a polar question, eyebrows are slightly raised, and the head tilted slightly forwards (De Vos et al., 2009). Examples (1) and (2) show the difference between a statement and a polar question, to exemplify that solely the addition of non-manuals can completely change the meaning of a sentence. The non-manual marker changes Statement (1) into Question (2).

(1) ***Statement***

IK AFWASSEN
I do.the.dishes

'I do the dishes'

(2) ***Polar Question***

IK AFWASSEN (accompanied by raised eyebrows and a head pushed slightly forward)
I do.the.dishes

'I do the dishes?' (accompanied with a raise in pitch)

The vital importance of non-manuals is also visible in negation. Regarding negative constructions, NGT is a non-manual dominant sign language. This means that the non-manual marker is more important than the manual one, and the latter is usually omitted (Baker et al., 2016). When a manual marker is used, however, this is usually for emphasis, which can be seen in Examples (3) and (4). A signer would use Sentence (3) to clarify that they are not going to the zoo tomorrow. They would use Sentence (4), however, to clarify the same point more sternly, by the addition of the sign NOT (Koolhof and Schermer, 2009). As a final note, the non-manuals discussed in the previous paragraphs are not an exhaustive list. Other non-manuals include (but are not limited to): a change of posture to fit the situation, noding the head for affirmation, or puffing the cheeks to indicate that the subject matter is large.

(3) ***Negative Construction***

MORGEN WIJ DIERENTUIN GAAN (with headshake)
tomorrow we zoo go

'We're not going to the zoo tomorrow.'

(4) ***Negative Construction with emphasis***

MORGEN WIJ NIET DIERENTUIN GAAN (with headshake)
tomorrow we not zoo go

'We're really not going to the zoo tomorrow.'

[1] A gloss consisting of multiple words separated by a hyphen denotes a single sign

(a) BOOK-FALLS (b) PERSON-FALLS

Figure 2.8: Classifiers for the verb TO-FALL in NGT (Zwitserlood, 2003)

Classifiers

In NGT (and other sign languages), the handshape of a sign may vary with different subjects. Figure 2.8 illustrates this with the NGT sign TO-FALL. In Figure 2.8a, the handshape is flat, resembling the book that is falling. Figure 2.8b, on the other hand, displays a handshape resembling the legs of a person, to indicate that a human is falling. This process is called 'classification', and is restricted to localisation in the signing space and verbs of motion (Baker et al., 2016). Classifiers are an important aspect of sign languages as they simplify conversation. A signer may localize a character with the person-classifier, and later on refer to this character by simply indicating the same classifier in the previously defined location.

2.2.3 Syntax

The syntax of Dutch and NGT are inherently different. Example (5) below demonstrates the difference in constituent order, adjective order, and verb conjugation. Firstly, NGT has a basic sentence order of Subject-Object-Verb (SOV), which indicates that the subject, object, and verb usually appear in that order (Baker et al., 2016). Dutch, on the other hand, has a sentence order of Subject-Verb-Object (SVO), switching around the object and the verb. Likewise, modifiers are deployed in a different order, such as the adjective succeeding the noun in Dutch, whereas the converse is usually true in NGT (Baker et al., 2016; Brunelli, 2011). Verb conjugation is not influenced by tenses, and only happens in the context of agreement (further explained in section 2.2.4). It only affects the manual component of the sign, the gloss and non-manuals do not change.

(5) [MAN OUD]$_S$ [HUIS]$_O$ [LOPEN]$_V$
'[De oude man]$_S$ [loopt naar]$_V$ [huis]$_O$.'
'The old man walks home.'

With negative and interrogative constructions, the location of the manual marker differs from the location of the respective Dutch word. A content question (shown in Example 6), has the WH-sign (signed counterpart of content question words) in the final sentence position. WH-doubling (the doubling of the WH-sign) is used often for emphasis, resulting in a WH-sign in initial and final sentence position (Baker et al., 2016). Furthermore, negative constructions (demonstrated in Example 7), have the manual marker positioned usually succeeding the subject, or at final sentence position (Oomen and Pfau, 2017).

(6) ***Content Question***

(WIE)	BEURT_HEBBEN	WIE
(who)	turn.has	who

'Whose turn is it?'

(7) ***Negation***

IK	NOOIT	BROCCOLI	ETEN	/	IK	BROCCOLI	ETEN	NOOIT
I	never	broccoli	eat	/	I	broccoli	eat	never

'I never eat broccoli.'

2.2.4 Signing Space

In sign language the signing space (Figure 2.2) is used to keep track of the conversation and make communication smoother. When localising entities in the signing space, the signer will do this logically and most likely based on their own perspective. This means that if the signer sees a tree to the right of a house, the signer will localise the tree to the right of the house in the signing space. By localising entities in the signing space, many aspects of communication are simplified (Baker et al., 2016).

During conversations, people will often reference something or someone mentioned before, instead of repeating it. It is unlikely that someone would say the following: ‘Bella is a young woman and Bella is pretty. Bella had blue earrings, but Bella lost the blue earrings.’. Instead, ‘Bella is a young woman and she is pretty. She had blue earrings, but she lost those.’ sounds much more natural. This is because of the use of personal and demonstrative pronouns, the former being words like ‘I’, ‘she’ and ‘it’, and the latter words such as ‘that’,‘those’ and ‘these’. In sign languages a similar method is used, but those pronouns are replaced by a pointing gesture, called an INDEX (Baker et al., 2016). The pointing gesture is made towards the place (locus) in the signing space that the entity being referred to has been localised or where they are in the surrounding space (e.g. for ‘I’ the signer simply points towards themselves). Figure 2.9 shows the loci that are most commonly used in sign language. Loci 1 and 2 are always present (the signer and the interlocutor), while loci $3a$ and $3b$ are used for referents that do not have to be present. So the sentence ‘I like you’ would be translated in sign language to INDEX$_1$ INDEX$_2$ LIKE. An INDEX can also be used to distinguish between far away (*there*) and nearby (*here*) locations. The former is performed by making a pointing gesture short and downward and the latter by making the gesture arc-shaped, longer and forwards.

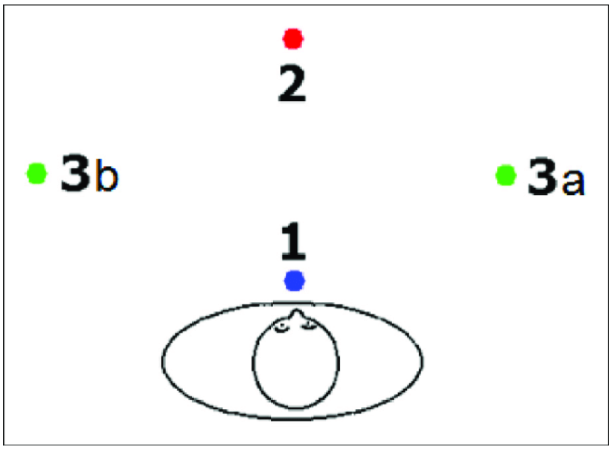


Figure 2.9: Indices in the signing space (Pfau et al., 2018)

Another important use of the signing space originates from the fact that in general, sign languages make little use of adpositions (prepositions and postpositions) to convey temporal (e.g. *before the meeting*), spatial (e.g. *on top of the house*) and abstract relations (e.g. *I did it for him*) between entities. Instead, these relations are mostly made clear using the signing space (Baker et al., 2016). For example, when signing ‘The boy walks to the cinema’, the sign for ‘walks’ is performed from the locus of BOY in the direction of the locus of CINEMA. So what is actually being signed is BOY INDEX$_{3a}$ CINEMA INDEX$_{3b}$ WALK$_{3a,3b}$, where the direction of movement and orientation of WALK makes the use of a sign for ‘to’ unnecessary. Whilst most sign languages have signs for expressing temporal relationships, the signing space can also be used instead. In NGT in order to sign ‘before the meeting’, a so-called ‘timeline’ is used. The two-handed sign for MEETING is made and then localised in the signing space by articulating INDEX with the non-dominant hand. The index finger of this hand remains at the locus of MEETING whilst the dominant hand moves from that location towards the signers body, indicating ‘before’. Had the movement been made in the other direction, so away from the signer’s body, then it would have meant ‘after the meeting’.

As shown earlier in the example of WALK, in sign language a verb might need to be modified to suit the sentence. Such verbs that are adapted to suit the subject and object, or rather their loci, are known as agreeing verbs (Baker et al., 2016). In the case of WALK its direction of movement and orientation are modified; in addition to the orientation of the sign, the start and end point of its movement indicate the subject and the object. Such verbs are known as directional verbs and allow the signer to identify the subject, verb, and object of a sentence with one gesture. For example, if the signer performed the sign for 'help' moving away from themselves, this would mean 'I help you' (HELP$_{1,2}$). However, if they had performed the same sign in the opposite direction, towards the signer, it would be 'Help me' (HELP$_{2,1}$). Directional verbs are not limited to involving the signer and the interlocutor, though. If the signer placed BOB to their right and HOME to their left and subsequently signed WALK from right to left, they would be signing BOB INDEX$_{3a}$ HOME INDEX$_{3b}$ WALK$_{3a,3b}$. However, not all agreeing verbs are directional verbs, as the orientation alone of a verb can also show agreement. For instance, CALL has no path movement in NGT and only has a small repeated movement from the wrist. In this case the orientation of the sign decides the subject and object. The back of the hand indicates the caller and the fingertips point towards whom they are calling out to. In the examples mentioned, the signs move from the subject of the sentence to the object and are orientated towards the latter. There are however also verbs where this is the other way around, such as the sign for INVITE in NGT. These verbs, whose target is the subject, are known as 'backwards verbs'.

2.3 Previous Research on Sign Language Translation and Synthesis

In the past two decades there has been increasing interest in the use of avatars for communicating in sign language as a result of a desire to make public spaces and services more accessible to the Deaf community (Wolfe et al., 2016). Avatars have the great advantage of allowing for more flexibility, as the signs being displayed can be easily changed. Furthermore, when combined with machine translation, the avatars can also be used for (automatic) translation between sign language and text or even speech, allowing for easier communication between hearing and Deaf people. For example, TESSA and her successor VANESSA were created in 2002 and 2004, to aid a Deaf person when they went to the post office, by translating the clerks speech into British Sign Language (Cox et al., 2002; Tryggvason, 2004). In another example, a system was developed that translated German train announcements into Swiss German Sign Language to be displayed on screens at train stations (Ebling and Glauert, 2013).

The use of sign language avatars is, however, not limited to the translation between spoken and sign language. PAULA is a computer-based sign language tutor that was created in 2006 to facilitate hearing adults in learning a limited vocabulary in American Sign Language (ASL) for use at the facility they all worked at (Davidson, 2006). The evaluation conducted with the staff on the use of PAULA showed that a sign language avatar can greatly improve the learning experience of hearing adults over other methods such as videos or even face-to-face lessons with a teacher. This is because the student can learn at their own pace and have the avatar repeat signs as little or as much as is necessary — and even adjust the speed at which the avatar signs. It is also much easier for the student return to a specific sign that they have difficulty with, and just repeat that one. More recent research in Geneva in 2017 lead to a similar conclusion (Rayner et al., 2017). Based on these conclusions, a sign language avatar would be an accessible and effective method for hearing adults to learn how to speak sign language. This would be particularly of use in situations where it is essential for a hearing person to learn to speak sign language.

Chapter 3

Hypothesis and Global Methodology

The background on NGT and the previous research into using an avatar to translate and synthesise sign language in the previous chapter lead to the hypothesis that an effective system consists of three fundamental components:

1. software for sign language synthesis;
2. lexical resources for encoding signs;
3. grammatical resources for structuring sentences

The following sections discuss each of these components in detail.

3.1 Avatar Software

When it comes to synthesising sign language, two research groups have produced the most promising results: PAULA and Java Avatar Signing (JASigning) (Davidson, 2006; Elliott et al., 2010). However, as the research for PAULA is very specific to ASL, and not open source, this project will utilise the JASigning software.

In order to animate sign language using JASigning, the signs must be encoded in Signing Gesture Markup Language (SiGML), which is an XML application. SiGML is based on the Hamburg Notation System for Sign Languages (HamNoSys).

3.1.1 The Hamburg Notation System

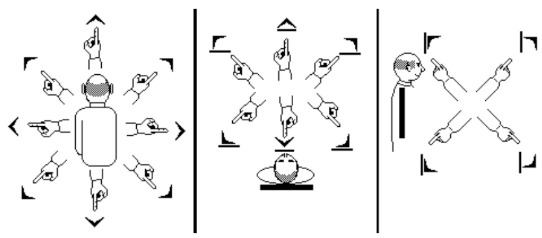
HamNoSys is an alphabetic system for transcribing signs using the five components of a sign: handshape, orientation, location, movement, and non-manuals, as described in sections 2.2.1 and 2.2.2. An example of the HamNoSys notation for HOUSE can be seen in Figure 3.1.

Figure 3.1: The HamNoSys Notation for HOUSE (Hanke, 2004)

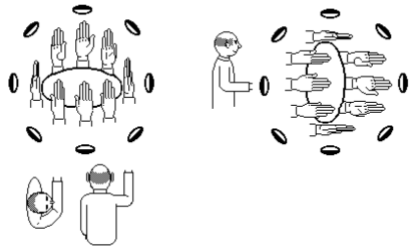
Handshape[1] is determined by the general shape of the hand (e.g. fist or open) and the position and optional bending of the thumb. Additionally, the position and optional bending of individual fingers can be specified (see Appendix A). Orientation describes the direction of the extended fingers[2] (or if they were to be extended) (see Figure 3.2a) and the direction of the palm[3] relative to them (see Figure 3.2b).

[1]The second symbol in Figure 3.1
[2]The third symbol in Figure 3.1
[3]The fourth symbol in Figure 3.1

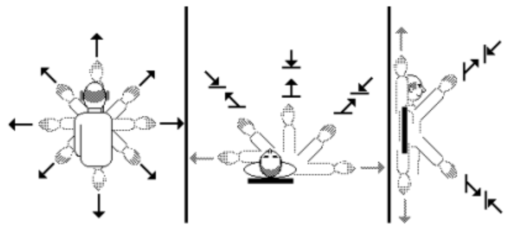
The location consists of two components, the first determines where in relation to the body[4] and the second[5] determines at what distance from the body the sign is performed (see Appendix B). In the case of two-handed signs, the location can also describe the relation of the two hands to each other. Actions describe in-place and path movements of hands[6], but can also describe the non-manual component of the sign (see Figure 3.2c). It can also be specified whether the actions are performed sequentially or simultaneously. Two-handed signs are indicated by the symmetry symbol[7] at the start of the description and specifying exceptions if the hands do not copy each other exactly (Hanke, 2004).



(a) Finger Orientations in HamNoSys



(b) Palm Orientations in HamNoSys



(c) Movements in HamNoSys

Figure 3.2: Orientations and movements in HamNoSys (Hanke, 2004)

The HamNoSys notation of a single sign starts with a description of the initial posture followed by the possible actions that are performed sequentially or simultaneously in order to change that posture. A posture consists of a description of the aforementioned components in order, meaning that the initial handshape is first, followed by the orientation of the fingers and palm, followed by an optional body part and relative location to it, followed by an optional movement (Hanke, 2004).

The great advantage of using HamNoSys is that it does not rely on the conventions of a sign language, as these differ from country to country. This means that HamNoSys can be used to describe any sign language, which is why it is one of the most widely used transcription systems. A disadvantage of HamNoSys, however, is that it mainly focuses on the manual components of a sign. The non-manual aspect is underdeveloped which means that the non-manual components of a sign cannot be controlled to the same extent as the manual components (Hanke, 2004).

[4] Omission signifies neutral space
[5] The fifth symbol in Figure 3.1
[6] The last two symbols in Figure 3.1
[7] The first symbol in Figure 3.1

3.1.2 Signing Gesture Markup Language

Signing Gesture Markup Language (SiGML) is an XML based language used for "generation of sign language performances by a computer-generated virtual human, or avatar" (Elliott et al., 2004).

There are two different types of SiGML: HamNoSys SiGML (H-SiGML) and gestural SiGML (G-SiGML). H-SiGML is based directly on HamNoSys, while G-SiGML is an extension of H-SiGML with more precise controls over the signing features. This project uses H-SiGML, as the database on which the corpus is based contains definitions of signs in this format. Listing 3.1 shows an example of SiGML code.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <sigml>
3
4 <hns_sign gloss="HUIS">
5   <hamnosys_nonmanual>
6     <hnm_mouthpicture picture="hYs"/>
7   </hamnosys_nonmanual>
8   <hamnosys_manual>
9     <hamsymmlr/>
10    <hamflathand/>
11    <hamextfingerul/>
12    <hampalmdr/>
13    <hamparbegin/>
14        <hamindexfinger/>
15        <hamfingertip/>
16        <hamplus/>
17        <hamindexfinger/>
18        <hamfingertip/>
19    <hamparend/>
20    <hamtouch/>
21    <hamshouldertop/>
22    <hamparbegin/>
23        <hammovedr/>
24        <hamsmallmod/>
25        <hamarcu/>
26        <hamreplace/>
27        <hamextfingero/>
28        <hampalml/>
29    <hamparend/>
30  </hamnosys_manual>
31 </hns_sign>
32
33 </sigml>
```

Listing 3.1: SiGML code for HOUSE

The start of a block of SiGML code is indicated by the notations in lines 1 and 2, and its end is indicated by the notation in line 33. Line 4 signals that the following lines encode the sign corresponding to the gloss. The non-manual component is given in lines 5–7, and in this case contains a mouth picture. The manual component is given in lines 7–30. As explained in the previous section, line 9 denotes that the left hand must mirror the right and that this is therefore a two-handed sign. Lines 10, 11 and 12 define the handshape, finger orientation and palm orientation. Lines 14–18 and 23–28 describe the movements in the sign, as signalled by the notations on lines 13 and 22. In the second movement the hands first move down at an angle, shaping the roof of the house, and then the wrists rotate so that the fingers face forwards, shaping the walls. The first movement, however, does not actually describe a movement. It defines the touching in line 20 as only pertaining to the index fingers and not the whole hands. Line 21 simply states the location at which the sign should be made. The result of this code can be seen in Figure 3.3.

Figure 3.3: HOUSE signed by an avatar

3.1.3 JASigning

JASigning accepts either H- or G-SiGML as input, but internally converts the former into the latter (John Glauert, personal correspondence, May 28^{th} 2020). Moreover, JASigning has additional functionality, including control of the duration of a sign, an option for a non-manual to over-arch onto multiple signs, or the addition of pauses between signs (Glauert and Elliott, 2011; Ebling and Glauert, 2013). JASigning is accessible via a website or an applet[8]. The user interface (UI) provides several options, which include changing the signing speed of the avatar, displaying the sign frame by frame, and showing the gloss of a sign. An image of the JASigning UI is displayed in Appendix C.

3.2 Lexical Resources

In spoken languages, there is often not merely one single translation for a word from one language to another. The same issue arises when translating between a spoken language and a sign language. An extra layer of difficulty is added to a translation in sign languages due to the use of classifiers (section 2.2.2). As a result, it is important to account for the fact that a word or concept should be represented in various ways in a lexicon. A system should be able to use the context of a word while computing its correct representation when multiple options are available. If, on the other hand, there are no known representations of a word, the system should still have the ability to convey the message of the sentence. In real life, signers often either improvise or fingerspell the word. As fingerspelling is less ambiguous, it is desirable for the program to resort to this rather than improvisation — which would not be an efficient way to learn the language. Lastly, the program should be able to count correctly, as learning to do so is an important aspect of studying a new language. All these resources should be contained in a database with encoded machine-readable signs.

[8]http://vhg.cmp.uea.ac.uk/tech/jas/vhg2020/index.html

3.3 Grammatical Resources

As mentioned in section 2.1, the signing space and non-manuals are essential for creating grammatically correct NGT sentences. Therefore, implementations of these two components are indispensable in a translator from glossed NGT to NGT. Through use of the signing space, sentences in NGT become much more comprehensible (section 2.2.4). The addition of non-manuals[9] to gestures makes signs comprehensible, but also increases the naturalness of sign language (section 2.2.2).

3.4 Overview Components

Figure 3.4 gives an overview of the expected final product and its components. This paper in particular will cover the implementation of the lexical processing. For an explanation of the implementation of the non-manuals and signing space see Corsel (2020) and Mende-Gillings (2020).

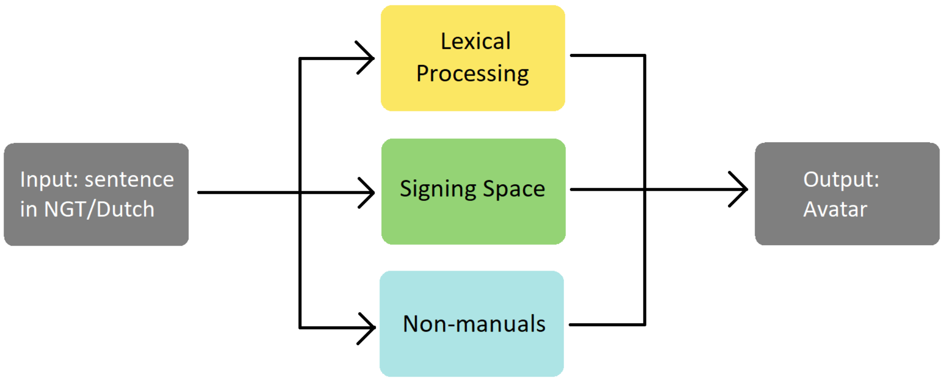


Figure 3.4: Outline necessary components

[9]Interesting to note is that previous research regarding sign language synthesis often neglects the implementation of non-manuals

Chapter 4

Lexical Resources

In order for an avatar to be able to make use of proper sign language, a solid foundation is necessary. As explained in section 3.2, this foundation consists of a corpus of signs, the correct use of classifiers, the ability to fingerspell where necessary, and the ability to sign numbers correctly. The goals for this specific project are therefore formulated as follows:

1. construct a database with a basic SiGML lexicon;
2. implement classifiers;
3. implement fingerspelling;
4. implement number signing

4.1 Approach

4.1.1 Database

The first goal of this project is to construct a database that contains a basic SiGML lexicon, which would form the basis of the program. Since the transcription of signs to HamNoSys (and SiGML) requires a considerable amount of effort, the best course of action is to make use of available resources. After contacting numerous researchers for assistance in obtaining an available database, Dr. Zwitserlood provided this project with two separate databases she created for Prins and Janssen (2014).

The first database is an Excel (.xlsx) file, and requires little cleaning. A collection of HamNoSys notations for English signs contained in the database is unnecessary for this implementation and thus removed. Subsequentially, the HamNoSys notation for every sign is translated into its SiGML equivalent. The second database is contained within a REALBasic Database file (.rbd), which requires Windows XP to open. However, the software that was previously used to open this filetype is currently too out of date, and as a result unable to open the database. As a resort, the .rbd file is opened with a text editor, resulting in code shown in Figure 4.1: randomly encoded with unknown characters. Through the use of regular expressions, an attempt at cleaning this file is made. Where possible, HamNoSys notations for signs are translated into SiGML.

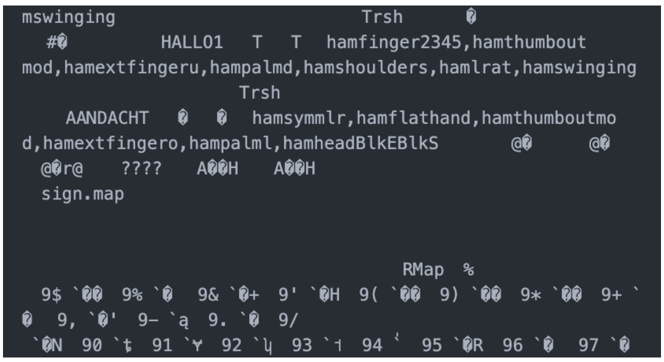


Figure 4.1: A sample of the .rbd file opened with a text editor

4.1.2 Classifiers

For some words, the database contains differing representations that may be used depending on the context. The program should take the rest of the sentence into account when computing which gloss is appropriate for a word given its context.

To implement this, a system is developed in which scores are assigned to each possible representation match (Table 4.1). The encoding of the gloss with the highest score total is then used in the translated sentence. As seen in the table, when a target word appears in the corpus as an exact match with an existing gloss it will be obtain a score of 3. If a word appears in a representation, the match will be given a score of 2. If a word in the context of the sentence appears in the beginning of a matching gloss, an additional score of 1.5 is given. If a context word is found at the end of a a matching gloss, 1.25 is added to the score. If a context word appears in brackets in the representation, an additional score of 1.25 is added to the total score.

The motivation behind these scores is best illustrated by an example input of MAN BOOK READ, with a target word of BOOK. If the corpus contains a gloss BOOK, a score of 3 will be given to the match. The gloss FULLY_BOOKED will also be a match, with a score of 2 for the appearance of BOOK. The gloss READ_BOOK will obtain a score of 3.5: 2 for BOOK, and 1.5 for the context word READ at the beginning of the gloss. If the previous gloss would be represented as either BOOK_READ or BOOK_(READ), the score would be 3.25. The glosses READ_BOOK (and its variations) are superior to the gloss BOOK, as they contain a context word. The gloss FULLY_BOOKED is inferior to BOOK, as it is safer to sign BOOK in case no gloss with context words is found.

Condition	Score
Word exactly the same as gloss	3
Word appears in gloss	2
Context word appears at the beginning of gloss	1.5
Context word appears at end of gloss	1.25
Context word appears in brackets in gloss	1.25

Table 4.1: Scoring system for matches

4.1.3 Fingerspelling

As the database lacks the manual alphabet in NGT[1], a translation of the letters to HamNoSys is necessary to create their SiGML representations. Listing 4.1 displays pseudocode to illustrate how a word is fingerspelled: the program first runs through a loop in which it checks whether a letter is single or double[2] and adds all letters to the 'letters' list (lines 5–15). After this, it loops through the 'letters' list and adds the SiGML code to the 'total_sigml' variable (lines 16–18).

```
1  def fingerspell(word):
2    letters = []
3    total_sigml = ''
4    num_letters = 1
5    for i in range(length(word)-1):
6      curr_letter = word[i]
7      if curr_letter == word[i+1]: # current letter same as next letter
8        num_letters += 1 # add to letter count
9      else: # current letter no longer equal to next letter
10       while num_letters > 2:
11         letters.append("".join([curr_letter,]*2)) # append joined double letters to list
12         num_letters = num_letters - 2 # update count
13       letters.append("".join([curr_letter,]*num_letters)) # add leftover letters to list
14       num_letters = 1 # reset letter count
15   letters.append(word[-1]) # append last letter to list
16   for letter in letters: # fingerspell each letter
17     letter += '_VINGERSPELLEN' # ensure that fingerspell entry gets found in corpus
18     total_sigml += get_sigml_neg(letter) # sign letter
19   return total_sigml
```

Listing 4.1: Pseudocode for fingerspelling a word

[1]Appendix D displays the manual alphabet of Sign language of the Netherlands
[2]Double letters are signed slightly differently

An example is illustrated by the word ‘haar’ (hair). The first ‘curr_letter’ is ‘h’, and since the following letter (‘a’) is not identical, the condition on line 9 is met. ‘Num_letters’ is not bigger than two, so lines 11 and 12 are ignored. The letter ‘h’ is appended to the ‘letters’ list as a single ‘h’, as ‘num_letters’ is one. The next letter, ‘a’, meets the condition on line 7, so ‘num_letters’ is increased to a value of 2. As the following ‘curr_letter’ is ‘a’, and its next letter is ‘r’, the condition of line 7 is not met. Again, ‘num_letters’ is not bigger than two, so lines 11 and 12 are ignored. The letter ‘a’ is appended to the ‘letters’ list as a double ‘a’ (aa), as ‘num_letters’ is two. The last letter is appended to the ‘letters’ list in line 15, which now contains the following elements:

```
letters = ['h', 'aa', 'r']
```

Listing 4.2: Current elements of ‘letters’

Finally, for each of these elements the SiGML code is added to ‘total_sigml’ in lines 16–18, which is returned with the information shown in Listing 4.3 below.

```
total_sigml = 'sigml_h_vingerspellen, sigml_aa_vingerspellen, sigml_r_vingerspellen'
```

Listing 4.3: Final SiGML

Lines 10 through 12 are to prevent that letters are combined as more than doubles (e.g. ‘aaa’), as there are existing no signs for this condition. In such a case, this code ensures that an entry as such would appear in the ‘letters’ list as:

```
letters = ['h', 'aa', 'a', 'r']
```

Listing 4.4: Letters list ‘haaar’

4.1.4 Number Signing

In NGT, signing numbers is quite simple: the patterns in the signs are similar to the patterns in Dutch counting. For example, the number 356 in Dutch would be pronounced as ‘driehonderd zes-en-vijftig’ (literal translation ‘three hundred six and fifty’). In NGT, one would sign this as the signs THREE HUNDRED SIX FIFTY, exactly the same as the Dutch pronunciation. Like fingerspelling, in order for the program to be able to sign numbers correctly, it requires a HamNoSys translation (and SiGML representation) for the numbers one to a hundred, and for a thousand, a million, and a billion.

```
def sign_number(number):
    total_sigml = ''
    numbers_to_sign = list(number)
    bignumber = {4:'THOUSAND', 5:'THOUSAND', 6:'THOUSAND', 7:'MILLION', 8:'MILLION',
        9:'MILLION', 10:'BILLION', 11:'BILLION', 12:'BILLION'}
    while numbers_to_sign is not empty:
        length = len(numbers_to_sign)
        modular = length%3
        if modular == 2: # the number is between 10-100
            sigml += get_sigml(numbers_to_sign[0] + numbers_to_sign[1]) # sign numbers
            numbers_to_sign = numbers_to_sign[2:] # remove signed numbers
        if modular%3 == 1: # the number is between 0-10
            sigml += get_sigml(numbers_to_sign[0])
            numbers_to_sign = numbers_to_sign[:1] # remove signed number
        if modular == 0: # the number is in the hundreds
            sigml += get_sigml(numbers_to_sign[0]) # sign how many hundred
            sigml += get_sigml('HONDERD') # sign ‘hundred'
            sigml += get_sigml( numbers_to_sign[1] + numbers_to_sign[2]) # sign tens
            numbers_to_sign = numbers_to_sign[3:] # remove signed number
        if length in bignumber.keys(): # thousands, millions, billions
            sigml += get_sigml(bignumber[length]) # sign ‘thousand/million/billion'
    return total_sigml
```

Listing 4.5: Pseudocode for signing numbers

Listing 4.5 displays pseudocode of how a number is signed, and is best illustrated through an arbitrary example number of '63007425'. In line 3, the program creates a list of the input number. Next, for as long as there are still numbers in the list, it enters a loop. First, the length of the list is stored in a variable (line 7), and the result of the length modulo three is calculated (line 8). The information stored in variables is now as follows:

```
numbers_to_sign = ['6', '3', '7', '9', '1', '4', '2', '5']
length = 8
modular = 2
total_sigml = ''
```

Listing 4.6: Current variable values

If the modular is 2, it means that the first number to sign is between 10 and 100 (as the 63 in the example number is). All these numbers have their own entry in the corpus, so the first and second elements of the list are merged together and the resulting SiGML code is added to the total 'total_sigml' variable (line 10). After this, the first two elements are removed from numbers_to_sign, since they are no longer necessary. Lines 12 through 19 are ignored, since their prerequisite conditions are not satisfied. In line 21, the previously calculated length of the list is compared against the keys in the 'bignumber' dictionary from line 4. If the length equals one of these keys, the SiGML code of its value is added to the 'total_sigml' variable. The length corresponds to the sign MILLION, so the loop is now repeated with the following information:

```
numbers_to_sign = ['7', '9', '1', '4', '2', '5']
length = 6
modular = 0
total_sigml = 'sigml_63, sigml_million'
```

Listing 4.7: Current variable values

The conditions of lines 9 through 14 are not met, and are thus ignored. The modular is 0, so the first element of the list (7), the sign HUNDRED, and the combination of the following two numbers (9 and 1) are added to the 'total_sigml' variable. After this, these three elements are removed from 'numbers_to_sign'. Again, the length is compared to the keys in the 'bignumber' dictionary, and THOUSAND is added to 'total_sigml'. The loop is repeated once more with these updated variables:

```
numbers_to_sign = ['4', '2', '5']
length = 3
modular = 0
total_sigml = 'sigml_63, sigml_million, sigml_7, sigml_hundred, sigml_91, sigml_thousand'
```

Listing 4.8: Current variable values

Again, the modular is 0 so the previous steps are repeated. This time, however, the condition on line 20 is not met, so no extra sign is added to the 'total_sigml' variable. As 'numbers_to_sign' is now empty, the loop is terminated, and the 'total_sigml' variable is returned with the information shown in Listing 4.9 below.

```
total_sigml = 'sigml_63, sigml_million, sigml_7, sigml_hundred, sigml_91, sigml_thousand,
     sigml_4, sigml_hundred, sigml_25'
```

Listing 4.9: Final SiGML

Note that, for the sake of simplicity, Listing 4.5 omits code for the following edge cases:

1. input such as '100', '1000', etc. will result in just the corresponding sign, HUNDRED, THOUSAND, in lieu of the superfluous ONE before this sign;
2. input such as '07', '007', '00', and '000' will ignore the zeroes;
3. input such as '007' or '017' will not generate the sign HUNDRED;
4. an input of '0' will generate the sign ZERO

Chapter 5

Results and Evaluation

In this chapter, the overall workings of the final system are first described in section 5.1.1, after which the results of the goals are described in section 5.1.2. Section 5.2 first evaluates the global results in section 5.2.1, and the lexical resources in section 5.2.2.

5.1 Results

5.1.1 Global Results

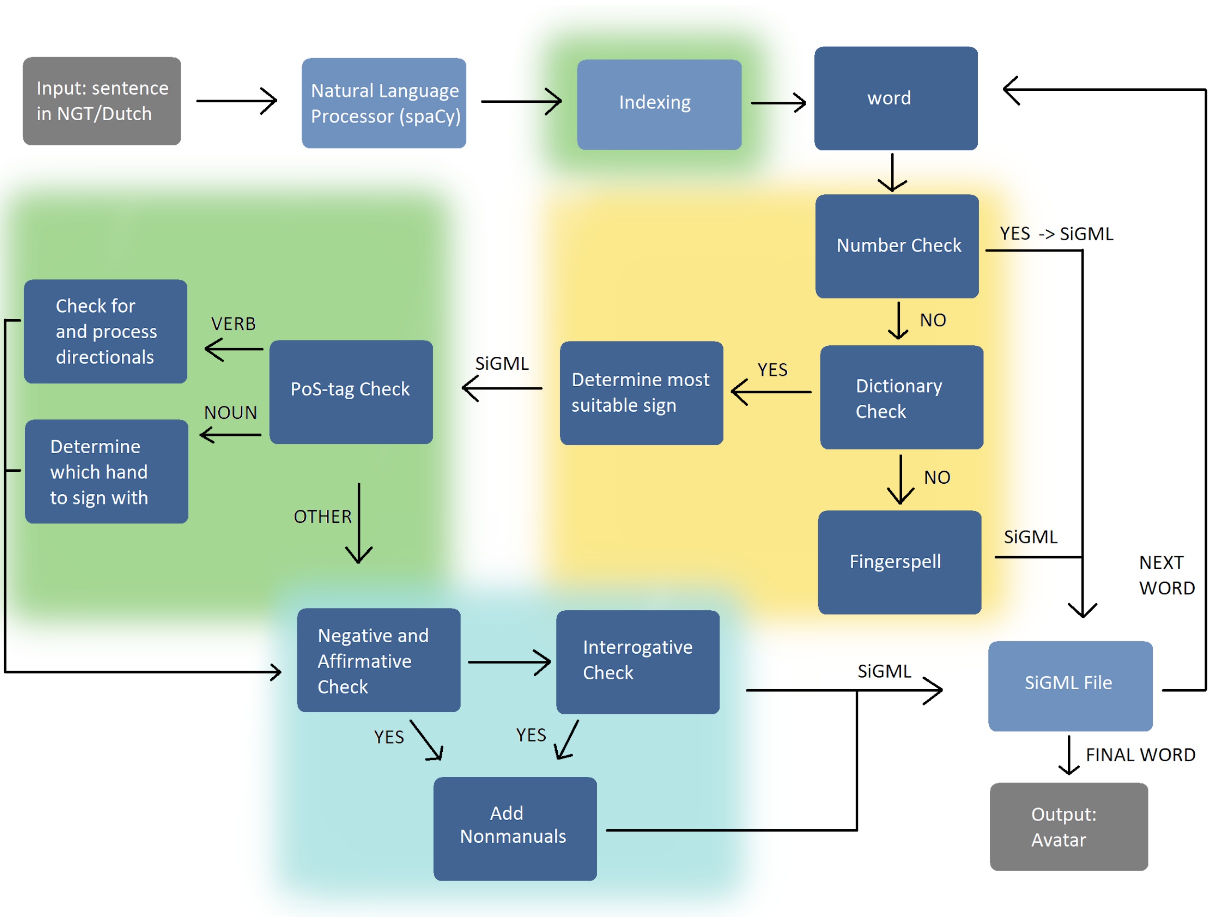


Figure 5.1: Pipeline of the entire program (the components part of lexical resources, the signing space, and non-manuals are indicated respectively by the yellow, green and light blue background squares)

The pipeline in Figure 5.1 shows the overall mechanism of the final system[1]. The program starts with an input sentence that can either be in NGT or an intermediate form of Dutch and NGT. It also accepts sentences in Dutch — as it produces signs in the order of the sentence — however, the output might not be in grammatically correct NGT. In addition, the program is not able to process multiple sentences at a time. Table 5.1 explains which input will produce correct versus incorrect output sentences. The first incorrect input sentence, MAN HUIS LOOPT, is incorrect due to the conjugation of LOPEN to LOOPT. The sentence MAN NAAR HUIS LOPEN is incorrect due to the word NAAR (to), since NGT makes little use of adpositions. Instead, indices may be used to indicate relations between entities, as can be seen in the second correct input sentence MAN INDEX$_{3A}$ HUIS INDEX$_{3B}$ LOPEN. Finally, the sentence DE MAN HUIS LOPEN contains the erroneous article DE, which should be omitted.

Correct Input	**Incorrect Input**
MAN HUIS LOPEN MAN INDEX$_{3A}$ HUIS INDEX$_{3B}$ LOPEN	MAN HUIS LOOPT MAN NAAR HUIS LOPEN DE MAN HUIS LOPEN

Table 5.1: Examples of correct and incorrect input of the Dutch sentence 'De man loopt naar huis' (The man walks home)

The program consists of three main steps: pre-processing, translation of words into SiGML, and communication with the avatar. The first step of the program is the pre-processing of the sentence. SpaCy[2] is a natural language processor used to acquire the Part-of-Speech (PoS) tags and dependencies of the words in the sentence. In the case that the sentence is not in NGT, indices are added where necessary. The next step is completed for each individual word in the sentence. In order to ensure that the avatar knows which signs to produce, words are assessed on type. The program first checks whether the current word is a number and, if so, adds its SiGML to the file. Next, in the case that the word is not in the dictionary (the corpus of signs), it will be fingerspelled by the avatar, and otherwise an algorithm is applied to determine which sign in the dictionary is the most suitable to use and retrieves SiGML. After the corresponding sign has been chosen, the PoS-tag of the word is evaluated. If the word is a noun, the program checks with which hand to sign the word, and if the word is a verb, the program checks whether it is a directional verb and thus needs to be adapted. The final check determines whether the sentence is interrogative, negative, or affirmative, so that the appropriate non-manuals can be added to the sign. Once the word has been processed, the sign is added to a file which collects the SiGML of all the words in the sentence. Finally, after every word in the sentence has been processed, the file is sent to the avatar which then produces the signs in order.

5.1.2 Results Lexical Resources

Database

The complete collection of signs encoded in the first database has been successfully retrieved and serves as a foundation for the corpus of signs used by the program. Unfortunately, it was not possible to salvage the second database in its entirety. Its 'encryption' occurred in different random locations, which caused an inability to use regular expressions in order to clean it up completely. Nevertheless, around fifty additional useful signs were retrieved through this method. The final corpus consisted of 2226 signs within a Python dictionary, formatted as in Listing 5.1. The keys consist of glosses, of which the value is contained in a tuple consisting of the SiGML code and the mouth picture for that sign.

```
Corpus = {
'DINSDAG':('hamflathand,hamextfingerl,hampalml,hamshoulders,hamtouch,hammoved,
            hamchest,hamtouch','dinst'),
'TWEE_2':('hamfinger23spread,hamthumbacrossmod,hamextfingeru,hampalmu,hamshoulders,
    hamlrat,hambetween,hamshoulders,hamclose','ve'),
'UUR':('hamfinger2,hamthumbacrossmod,hamextfingeru,hambetween,hamextfingerul,
        hampalml,hamneck,hamclose,hamcirclei,hamclockr','u:r')
}
```

Listing 5.1: Formatting of the corpus

[1]For instructions on how to install and use the program, see: https://github.com/LykeEsselink/SignLanguageSynthesis
[2]https://spacy.io

Sentence	Matches	Score
‘PAARD HOUDEN VAN SPRINGEN’	SPRINGEN_KIKKER	2
	SPRINGEN_PAARD	3.25
‘JONGEN BOOS OMDAT BRUG OPEN’	AFGELOPEN_DAGEN	2
	GEOPEND_ZIJN	2
	OPEN	3
	OPEN_ZIJN	2
	OPEN_ZIJN_VAN_BRUG	3.25
	OPEN_ZIJN_VAN_BOEK	2
	OPENDOEN_VAN_DEUR	2
	UITPAKKEN_/_OPENMAKEN	2

Table 5.2: Scores for classifier matches of test sentences

Classifiers

Table 5.2 illustrates how scores are assigned the matches found by the program for test sentences ‘PAARD HOUDEN VAN SPRINGEN’ (the horse loves to jump) and ‘JONGEN BOOS OMDAT BRUG OPEN’ (the boy is angry because the brigde is open). For the first sentence, two matches are found for the word ‘springen’: the classifier for a jumping frog, and the classifier for a jumping horse. As expected, the latter is chosen. Furthermore, a selection of matches the word ‘open’ is shown in the Table. Matches that begin with OPEN_ZIJN (to be open) are assigned higher scores than the other possible matches (with the exception of OPEN). In line with the expected outcome, the sign OPEN_ZIJN_VAN_EEN_BRUG (a bridge being open) has the highest score, and is selected by the program as the best match.

Fingerspelling

As seen in Figure 5.2, the avatar has the ability to fingerspell. With every letter, her mouth performs the mouth sign for that letter. She is unable to combine mouth signs for vowels such as ‘au’, ‘eu’, ‘oe’, ‘ui’, and ‘ie’, but is able to combine the mouth sign for double letters such as ‘aa’, ‘ee’, and so forth. However, not all signs are completely correct. The thumb position for the letter ‘e’ is placed next to the fingers, while it should be in the fist. Moreover, the letter ‘p’ can be mistaken for the letter ‘d’, as the shape of the fingers is slightly off. Lastly, the movement of the letter ‘j’ is faulty: the avatar first signs the letter ‘i’ and pauses before moving the wrist in a swooping motion. The correct sign consists exclusively of the swooping motion.

Figure 5.2: The avatar fingerspelling GEBARENTAAL (sign language)

Figure 5.3: The avatar signing the number '3571'

Number signing

Figure 5.3 illustrates how the avatar signs a number. This is done in the same order as NGT number signing conventions. As with the fingerspelling, the mouth sign for every number is coded into the SiGML representation for each number. Like the manual signs, the mouth signs are performed one after the other (in combination with the manual signs), so there is currently no possibility to increase the flow of how the mouth signs fit together.

5.2 Evaluation

5.2.1 Global Evaluation

In order to assess the overall performance of the program, eighteen test sentences (see Appendix E) were constructed and given to the program to sign in NGT. The output of the program — the signing performed by the avatar — was recorded for each sentence and stored in a database along with its meaning. Two evaluators, one native speaker of NGT and one who learned it as an adult, were asked to watch the videos and fill out an evaluation form (Appendix F). The first step of the evaluation required them to interpret the meaning of the sentence based on the signing of the avatar. In the second step they were shown what the avatar was meant to sign, followed by various questions to assess the comprehensibility of the sentence and the naturalness of the signs.

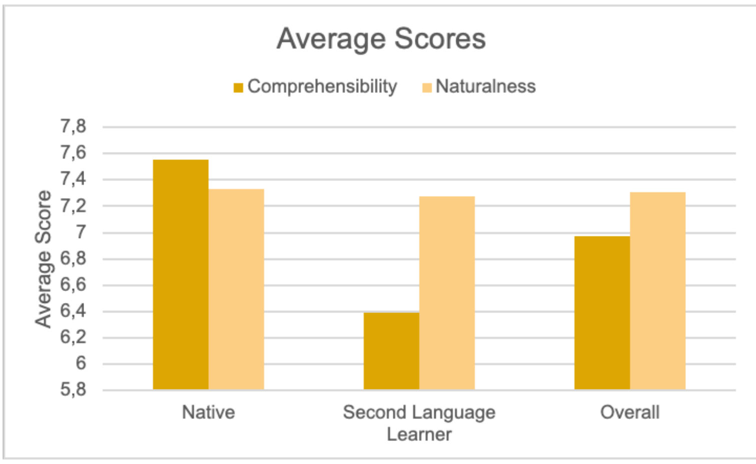


Figure 5.4: Average scores of comprehensibility and naturalness

The evaluators were asked to score the comprehensibility and naturalness of each sentence on a scale from 1 to 10. Figure 5.4 shows the average results of the scoring [3]. Overall, the naturalness scores higher than the comprehensibility, with average scores of 7.31 and 6.97 respectively. Furthermore, the average scores given by the native speaker, 7.33 and 7.56, are slightly higher than the average scores given by the non-native speaker, 7.28 and 6.38, for respectively naturalness and comprehensibility. It is important to note that the feedback revealed that not all signs in the database are correct, which influenced the scores considerably.

[3] Appendix G shows the overall results of the scoring

	NGT	Dutch
1	MAN 33 PLANT KOPEN	De man koopt 33 planten
2	VANAVOND KIND WIL SNOEPJES ETEN	Het kind wilt vanavond snoepjes eten
3	PAARD HOUDEN VAN SPRINGEN	Het paard houdt van springen
4	MEISJE TENTAMEN WILLEN HALEN	Het meisje wilt het tentamen graag halen
5	JONGEN BOOS BRUG OPEN	De jongen is boos want de brug staat open
6	KASTEEL 3571 JAAR OUD	Het kasteel is 3571 jaar oud

Table 5.3: Test sentences provided for evaluation of lexical resources

5.2.2 Evaluation Lexical Resources

For the evaluation of the lexical resources, only the test sentences shown in Table 5.3 are considered. These contain: two sentences pertaining to correct use of classifiers (sentences 3 and 5), two sentences pertaining to fingerspelling (sentences 2 and 4), and two sentences pertaining to signing numbers (sentences 1 and 6).

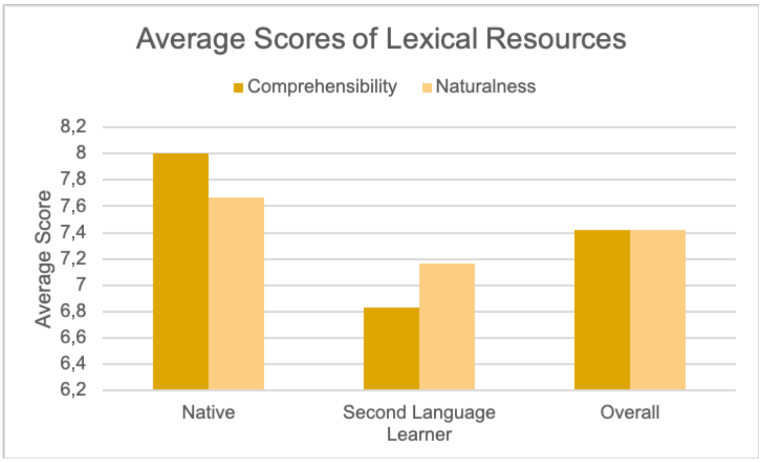


Figure 5.5: Average scores of lexical resources comprehensibility and naturalness

Figure 5.5 shows the average results of the scoring given to the comprehensibility and naturalness of the sentences in Table 5.3. In line with the trend in section 5.2.1, the native speaker assigned higher overall scores to both comprehensibility (8) and naturalness (7.67) than the second language learner (6.83 and 7.17, respectively). However, the overall average score for both categories was identical at 7.42 — slightly higher than the overall average scores in the previous section.

According to the feedback given by the evaluators, not all signs in the database were encoded correctly. For example, the sign THOUSAND was encoded with a movement in the direction of the interlocutor instead of sideways, and the sign WOMAN was located at shoulder- instead of ear height. Furthermore, although the program was able to identify the correct classifiers for the test sentences, not all chosen words were correct. In test sentence 3, the sign HOUDEN_VAN was repeated, as the input sentence contained a space between 'houden' and 'van'. Both evaluators agreed that the fingerspelling was not consistently clear: some letters could be confused for others (e.g. the p for a d), and other letters had a wrong handshape or movement. The number signing, on the other hand, was evaluated to be quite comprehensible: the manual components were clear and correct. However, the evaluators found the mouth signs during number signing confusing, as they did not flow together naturally.

Chapter 6

Conclusion and Discussion

6.1 Conclusion

This project has taken steps in a direction of building an avatar that synthesises Sign Language of the Netherlands. Initially, the goal was to create a proof of concept that could use its limited vocabulary to sign basic sentences in NGT. The main research question was: "What are the necessary components for a system that uses an avatar to translate a Dutch sentence to Sign Language of the Netherlands?".

In the theoretical foundation, the history and grammar of NGT have been discussed, alongside previous research on sign language translation and synthesis. From this previous research, three fundamental components were established to answer the research question: software for sign language synthesis, lexical resources for encoding signs, and grammatical resources for structuring sentences. For the avatar software, this project made use of the Java Avatar Signing software, which required SiGML (a signing markup language based on HamNoSys). The lexical resources for encoding signs were discussed in chapter 4, and consisted of four steps: constructing a database with a basic SiGML lexicon, implementing the use of classifiers, fingerspelling, and the correct way of signing numbers. For the discussion of differing grammatical resources for structuring sentences, the reader is referred to the papers of Corsel (2020) and Mende-Gillings (2020).

The overall workings of the program were discussed in chapter 5, which also included a link to download and use the final product. Two evaluators (one native signer and one that learned NGT as an adult) scored eighteen sentences on their comprehensibility and naturalness. Overall, the native signer assigned relatively higher scores to both categories. Furthermore, chapter 5 contained the lexical resources results: most of the databases was used as a corpus, correct classifiers were chosen by the program, and both fingerspelling and the signing of numbers were completely implemented. The average comprehensibility and naturalness scores for the lexical resources were slightly above average, again with the native signer scoring both categories significantly higher. The evaluators agreed that some signs were encoded incorrectly in the database, and that fingerspelling was not always clear. They fully understood the signing of numbers, although the mouth signs accompanying this was seen as unnatural.

6.2 Discussion

Although a foundation has been laid for the translation and synthesis of Dutch to NGT, there are a number of points of improvement. First and foremost is that the database should be reviewed and corrected. The evaluators discovered a number flaws in a selection of the signs they were requested to evaluate (some signs were produced at wrong locations or with the wrong movement, handshape, or orientation). This needs to be addressed, and would improve the comprehensibility of the signs synthesised by the avatar. Furthermore, the facial expressions of the avatar are currently extremely neutral. In NGT, facial expressions play a crucial part in the meaning of a sign. To further improve comprehensibility, future research should include this vital aspect as a goal. Additionally, the corpus currently consists of solely the manual sign and the mouth picture, neglecting non-manuals such as the previously mentioned facial expressions, and other actions such as raised shoulders or a correct gaze. It would greatly improve both comprehensibility and the naturalness of signs were these non-manuals to be included in the corpus.

Furthermore, there are a number of issues that should be addressed pertaining to the lexical resources. Although the classifiers are computed correctly in a number of tested sentences, in some cases, selection of the correct representation for a word is not ideal. An example of this was given by the test sentence PAARD HOUDEN VAN SPRINGEN (the horse loves to jump) in section 5.1.2. Due to the word 'van', the final output sentence was PAARD HOUDEN_VAN HOUDEN_VAN SPRINGEN, as the program attempts to find the optimal word for each input word. To optimize this, a function should be written in which the program is able to recognize that the 'van' in this sentence pertains to 'houden', and is therefore not necessary to translate. Furthermore, instead of being based on a scoring system, the selection of classifiers should ideally be parameterised. This way, more sentences can be signed with the correct classification even if the corpus does not contain a specific representation for all possible classifiers. If this is not possible, an additional selection of classifiers should be added to the corpus so that the program has a more complete overview of possible representations of a word in every context. Similar to the classifiers, fingerspelling requires additional work in order to function at optimal effectiveness. As a number of handshapes are slightly incorrect, the SiGML encodings of these signs should be revisited. It is, however, unclear whether this is possible with H-SiGML. If this is not the case, it would be beneficial to translate the current corpus to G-SiGML so that more precise control of the signing features is available. Finally, the signing of numbers would benefit from a function that can calculate the mouth sign based on the complete number, which should then be layered over one complete sign in lieu of separate partitions. This would provide a more natural impression of the translation.

Overall, the results exceeded the expectations and goals set at the beginning of this project: the final system had access to a much larger corpus and had more advanced features than originally believed to be possible. The most important features such as correct phonology, non-manual gestures, and directional verbs were largely implemented. Nevertheless, it is important to note that this technology can- and should not replace human interpreters, as they are able to provide the expressive aspect of signing in ways that an animated avatar (currently) cannot. Additionally, an avatar will only sign one specific way and in one dialect, which could impede comprehensibility for Deaf people from differing regions. Moreover, the translation of text to signs currently takes a few seconds to compute, which is undesirable in situations where rapid and detailed communication or translation is required, such as press conferences.

Currently, the developed system is quite advanced: functionalities like the correct use of classifiers and non-manuals are usually not implemented in similar systems. This is not without reason: correct implementation is a tedious process and contains many layers. The current system is still a long way off from being entirely grammatically correct, and it will take a lot of effort to make this so. However, if successful, this system could prove an extremely useful tool for people attempting to learn NGT. Moreover, it can be altered for other purposes: simple announcements can be made at events, train stations, or airports. Alternatively, it can be used as a simple communication method between a hearing and Deaf person in casual conversation. Further research could allow communication between a (hearing) doctor and (Deaf) patient. The latter possibility is the next course of action for this research.

Bibliography

A. Baker, B. van den Bogaerde, R. Pfau, and T. Schermer. *The linguistics of sign languages: An introduction*. John Benjamins Publishing Company, 2016.

S. Brinkhuijsen. Interlingual machine translation for dutch to sign language of the netherlands, 2019.

M. Brunelli. *Antisymmetry and Sign Languages: A comparison between NGT and LIS*. LOT, 2011. ISBN 9789460930676.

A. J. Corsel. Multidimensionality in sign language synthesis: Translation of dutch into sign language of the netherlands, 2020.

S. Cox, M. Lincoln, J. Tryggvason, M. Nakisa, M. Wells, M. Tutt, and S. Abbott. Tessa, a system to aid communication with deaf people. In *Proceedings of the fifth international ACM conference on Assistive technologies*, pages 205–212, 2002.

M. J. Davidson. Paula: A computer-based sign language tutor for hearing adults. In *Intelligent Tutoring Systems 2006 Workshop on Teaching with Robots, Agents, and Natural Language Processing*, pages 66–72, 2006.

L. de Groot. Initiatiefwet: Nederlandse gebarentaal erkennen als officiële taal. *Nederlandse Omroep Stichting (NOS)*, 2019. URL `https://nos.nl/nieuwsuur/artikel/2300498-initiatiefwet-nederlandse-gebarentaal-erkennen-als-officiele-taal.html`.

C. De Vos, E. Van Der Kooij, and O. Crasborn. Mixed signals: Combining linguistic and affective functions of eyebrows in questions in sign language of the netherlands. *Language and Speech*, 52(2-3): 315–339, 2009. ISSN 00238309.

S. Ebling and J. Glauert. Exploiting the full potential of jasigning to build an avatar signing train announcements. (October), 2013.

R. Elliott, J. Glauert, V. Jennings, and J. Kennaway. An overview of the sigml notation and sigml-signing software system. In *Sign Language Processing Satellite Workshop of the Fourth International Conference on Language Resources and Evaluation, LREC 2004*, pages 98–104, 2004.

R. Elliott, J. Bueno, R. Kennaway, and J. Glauert. Towards the integrationn of synthetic sl annimation with avatars into corpus annotation tools. In *4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies, Valletta, Malta*, page 29, 2010.

K. Emmorey. *Language, cognition, and the brain: Insights from sign language research*. Psychology Press, 2001.

N. Gebarencentrum. Mini gebarenwoordenboek, 2020.

J. Glauert and R. Elliott. Extending the sigml notation – a progress report. *Second International Workshop on Sign Language Translation and Avatar Technology (SLTAT)*, 2011. URL `http://vhg.cmp.uea.ac.uk/demo/SLTAT2011Dundee/`.

T. Hanke. Hamnosys-representing sign language data in language resources and language processing contexts. In *LREC*, volume 4, pages 1–6, 2004.

J. A. Holt. Stanford achievement test—8th edition: Reading comprehension subgroup results. *American Annals of the Deaf*, 138(2):172–175, 1993.

W. A. Hoover and P. B. Gough. The simple view of reading. *Reading and writing*, 2(2):127–160, 1990.

J. Kennaway, J. R. Glauert, and I. Zwitserlood. Providing signed content on the internet by synthesized animation. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 14(3):15–es, 2007.

C. Koolhof and T. Schermer. *Van Dale Basiswoordenboek Nederlandse Gebarentaal*. 2009.

M. Marschark. Language development in children who are deaf: A research synthesis. 2001.

S. E. Mende-Gillings. The signing space for the synthesis of directional verbs in ngt, 2020.

G. Morgan and B. Woll. *Directions in sign language acquisition*, volume 2. John Benjamins Publishing, 2002.

S. Morrissey and A. Way. An example-based approach to translating sign language. 2005.

M. Oomen and R. Pfau. Signing not (or not): A typological perspective on standard negation in sign language of the netherlands. *Linguistic Typology*, 21(1):1–51, 2017. ISSN 1613415X.

R. Pfau, M. Salzmann, and M. Steinbach. The syntax of sign language agreement: Common ingredients, but unusual recipe. *Glossa: a journal of general linguistics*, 3(1), 2018.

M. Prins and J. Janssen. Automated sign language, 2014. URL `https://www.tno.nl/media/3385/automated_sign_language_tno_2014_r10825.pdf`.

M. Rayner, I. Strasly, N. Tsourakis, J. Gerlach, and P. Bouillon. Menusigne: A serious game for learning sign language grammar. pages 181–186, 2017.

T. Schermer and C. Koolhoof. *De Nederlandse Gebarentaal*. Van Tricht Twello, The Netherlands, 1991.

B. Smit. Rule-based machine translation from dutch to dutch sign language using the grammatical framework, 2019.

J. Tryggvason. Vanessa: A system for council information centre assistants to communicate using sign language. *School of Computing Science, University of East Anglia*, 2004.

M. Verlinden and I. Zwitserlood. Cki en gebarentaal. slideplayer.nl, 2016. URL `https://slideplayer.nl/slide/11272197/`.

L. N. Wauters, W. H. Van Bon, and A. E. Tellings. Reading comprehension of dutch deaf children. *Reading and writing*, 19(1):49–76, 2006.

S. D. Weille. Automatic translation of dutch into dutch sign language with a focus on the use of tense, 2019.

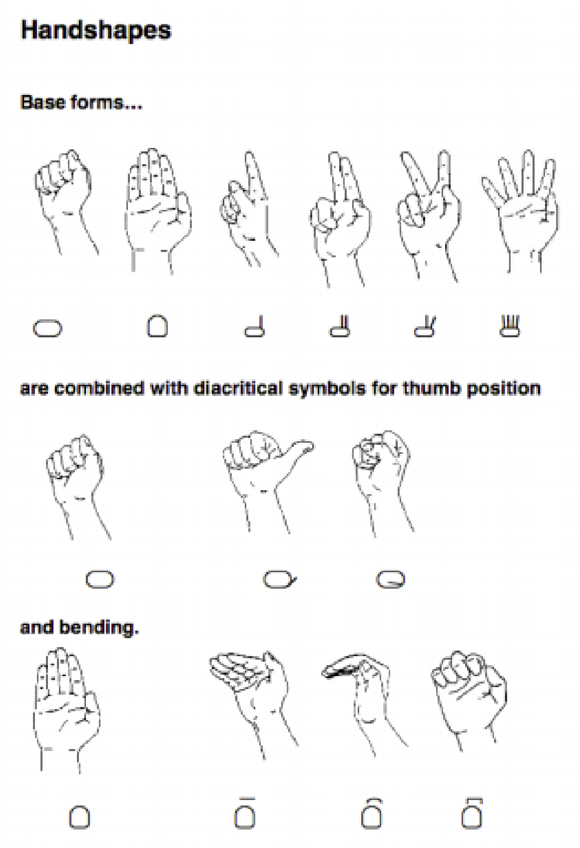
R. Wolfe, E. Efthimiou, J. Glauert, T. Hanke, J. McDonald, and J. Schnepp. recent advances in sign language translation and avatar technology, 2016.

I. E. P. Zwitserlood. *Classifying hand configurations in Nederlandse Gebarentaal (Sign Language of the Netherlands)*. PhD thesis, 2003.

Appendices

Appendix A

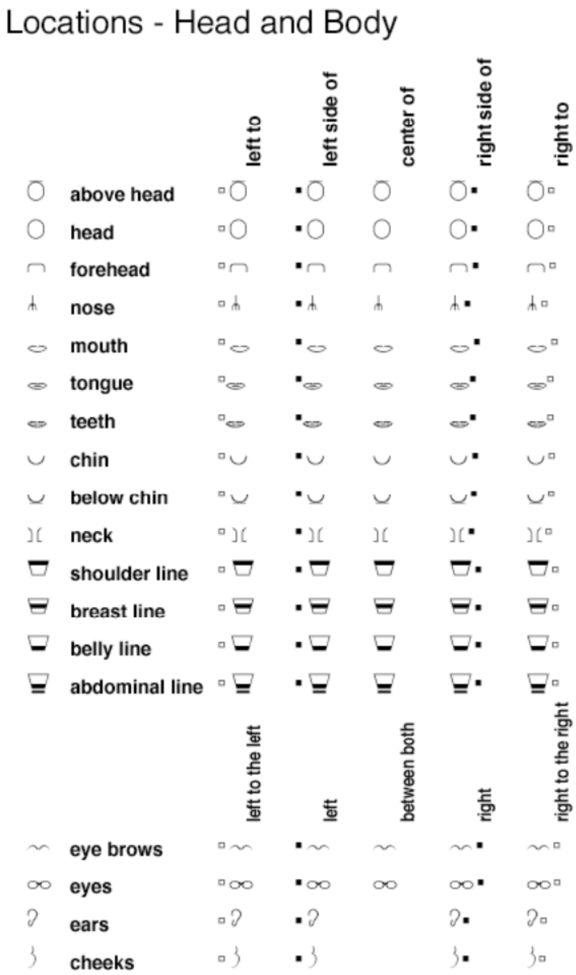
Hamnosys Handshapes



Hanke (2004)

Appendix B

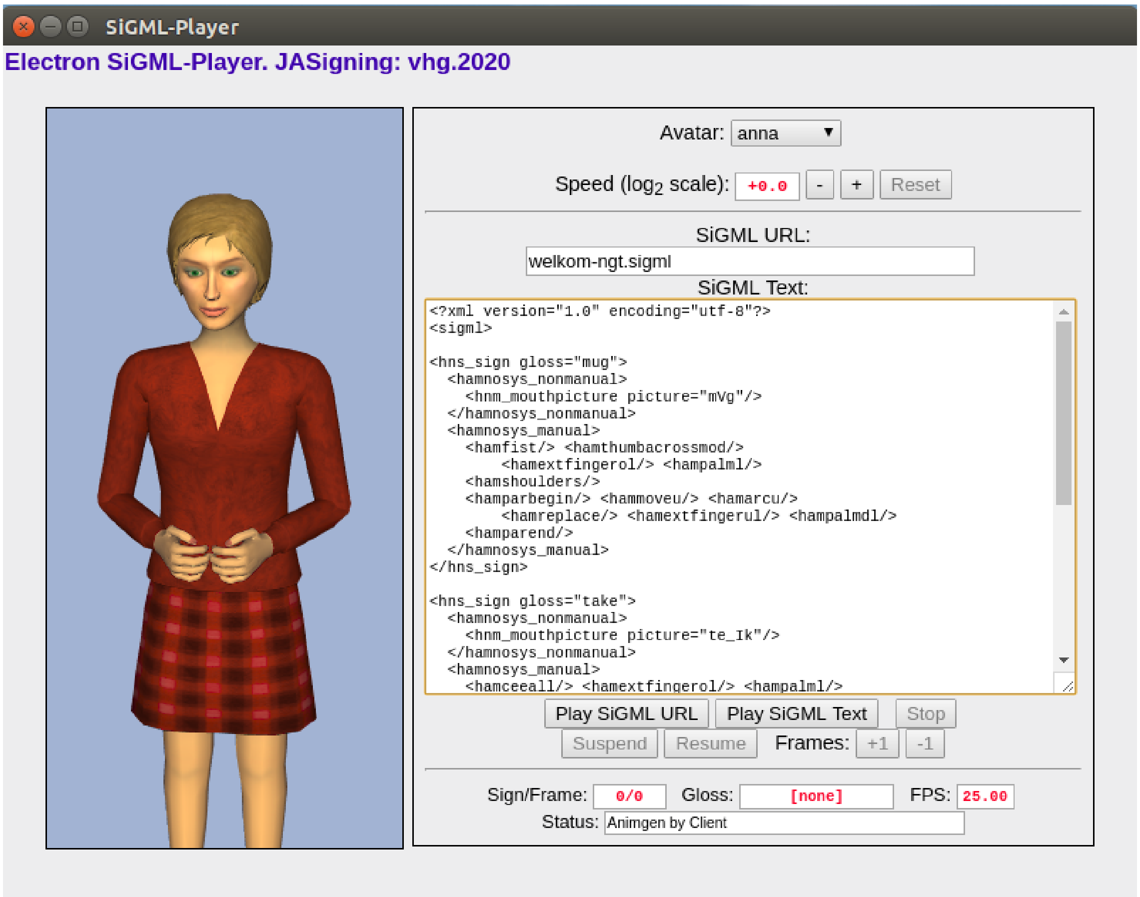
HamNoSys Locations



Hanke (2004)

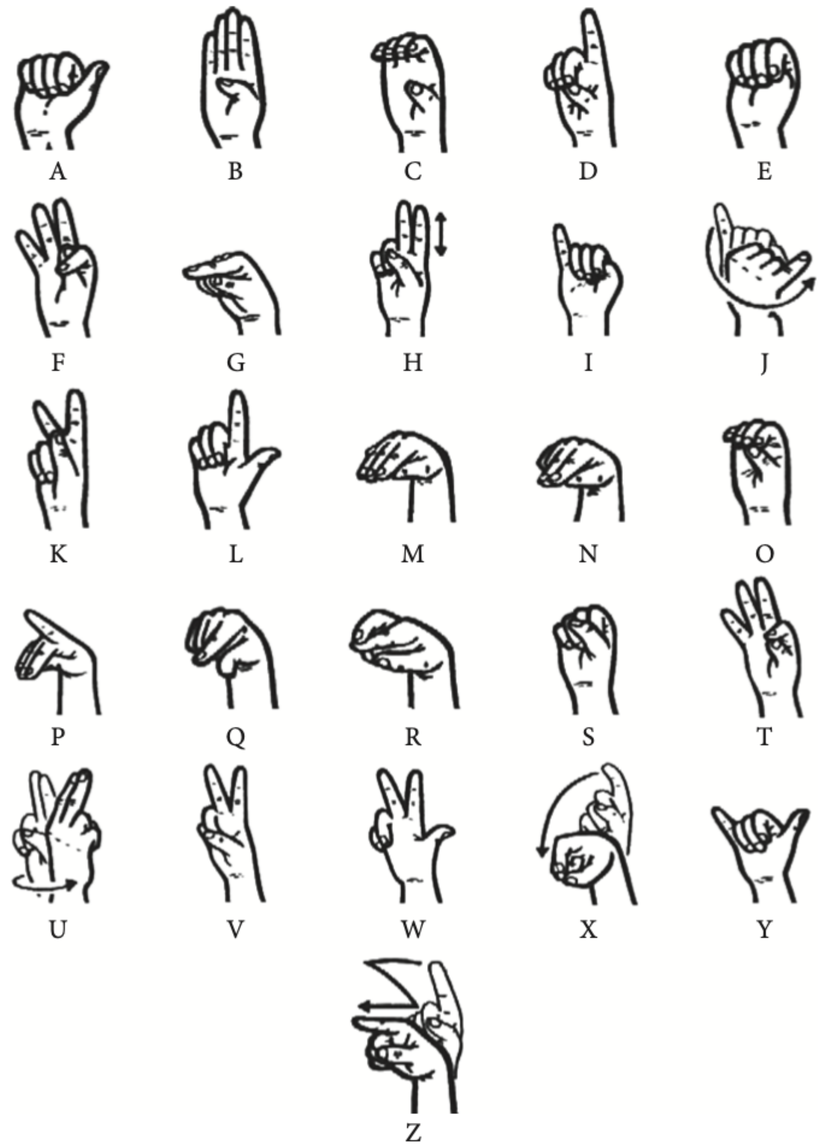
Appendix C

JASigning User Interface



Appendix D

The manual alphabet in NGT



Baker et al. (2016)

Appendix E

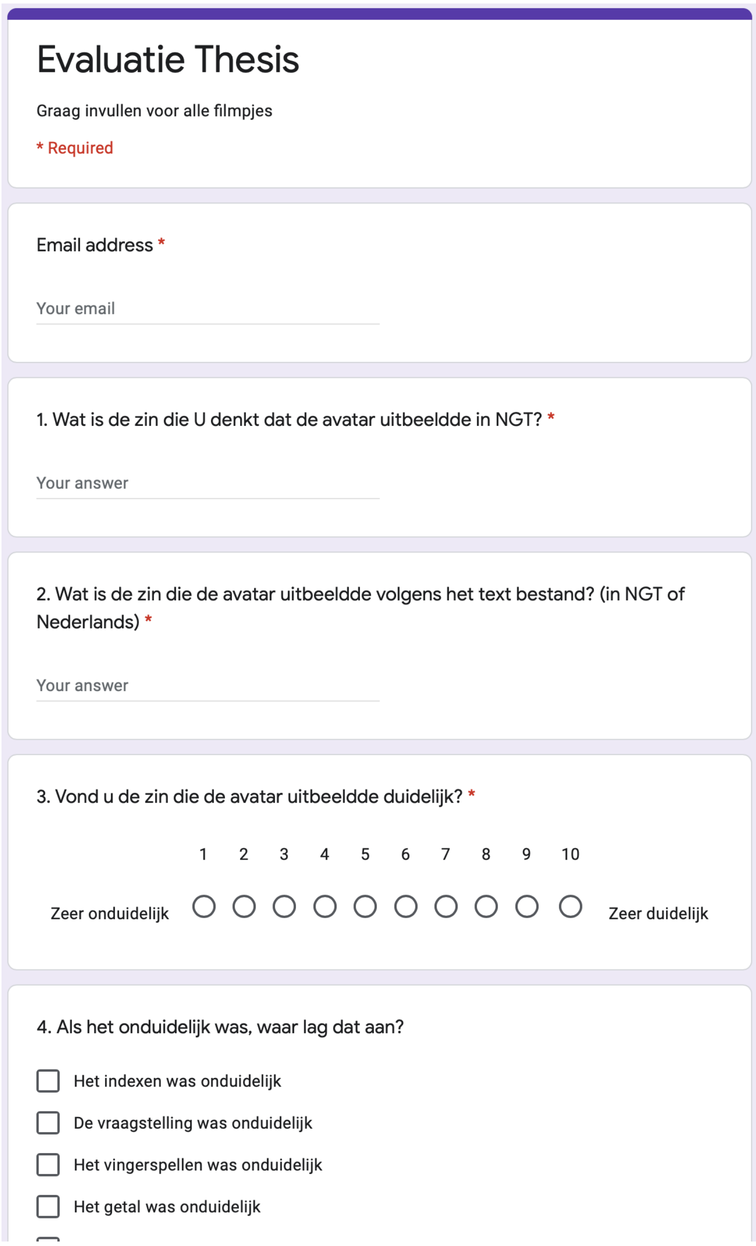
Evaluation Sentences

	NGT	Dutch
Lexical Resources	MAN 33 PLANT KOPEN VANAVOND KIND WIL SNOEPJES ETEN PAARD HOUDEN-VAN SPRINGEN MEISJE TENTAMEN WILLEN HALEN JONGEN BOOS BRUG OPEN KASTEEL 3571 JAAR OUD	De man koopt 33 planten Het kind wilt vanavond snoepjes eten Het paard houdt van springen Het meisje wilt het tentamen graag halen De jongen is boos want de brug staat open Het kasteel is 3571 jaar oud
Directional Verbs and Indexing	MAN LANG OUD HUIS MOOI GROOT KIJKEN IK OUD MAN LOPEN JIJ INDEX_3A GEVEN VROUW MOOI MANDARIJN HOUDEN VAN OP_AUX VROUW LANG OUD ZIJN VERDRIETIG MEISJE MOOI LANG VROUW MOOI BANG	De lange oude man kijkt naar het mooie grote huis Oude ik loopt naar de man Jij geeft aan haar/hem De mooie vrouw houdt van de mandarijn De lange oude vrouw is verdrietig Een mooi lang meisje en een mooie bange vrouw
Non-manuals	OUD-EN-NIEUW OLIEBOL ETEN WIE? VRACHTWAGEN BOTSEN WAAROM? NEEF NACHTMERRIE HEBBEN? JIJ 3 OF MEER DOCHTER HEBBEN? KEIZER LOPEN OF RENNEN? BOER NOOIT MELK VIES OF NOOIT EI VIES	Wie eet een oliebol op oud-en-nieuw? Waarom botste de vrachtwagen? Heeft de neef een nachtmerrie? Heb jij 3 of meer dochters? Loopt de keizer of rent de keizer? Heeft de boer nooit vieze melk, of nooit vieze eieren?

Table E.1: Test sentences provided for evaluation

Appendix F

Evaluation Form



Evaluatie Thesis

Graag invullen voor alle filmpjes

* Required

Email address *

Your email

1. Wat is de zin die U denkt dat de avatar uitbeeldde in NGT? *

Your answer

2. Wat is de zin die de avatar uitbeeldde volgens het text bestand? (in NGT of Nederlands) *

Your answer

3. Vond u de zin die de avatar uitbeeldde duidelijk? *

	1	2	3	4	5	6	7	8	9	10	
Zeer onduidelijk	○	○	○	○	○	○	○	○	○	○	Zeer duidelijk

4. Als het onduidelijk was, waar lag dat aan?

- [ ] Het indexen was onduidelijk
- [ ] De vraagstelling was onduidelijk
- [ ] Het vingerspellen was onduidelijk
- [ ] Het getal was onduidelijk

Figure F.1: Evaluation Form

- [ ] Het klopte grammaticaal niet
- [ ] Er was een gebaar fout
- [ ] Het gebaar was niet goed te zien
- [ ] Er zat een gebaar in de zin die er niet in hoorde
- [ ] Other:

5. Vond u de zin die de avatar uitbeeldde natuurlijk? *

	1	2	3	4	5	6	7	8	9	10	
Zeer onnnatuurlijk	○	○	○	○	○	○	○	○	○	○	Zeer natuurlijk

6. Als het onnatuurlijk was, waar lag dat aan?

- [ ] De handvorm was onnatuurlijk
- [ ] De locatie was onnatuurlijk
- [ ] De nonmanuele component was onnatuurlijk
- [ ] Het klopte grammaticaal niet
- [ ] De gebaren liepen niet goed in elkaar over
- [ ] De pauzes tussen de gebaren waren te lang
- [ ] Other:

7. Heeft u nog andere feedback voor deze zin?

Your answer

Submit

Page 1 of 1

Figure F.2: Evaluation Form

Appendix G

Global Results

Two figures showing the grades of comprehensibility and naturalness of the avatar.[1]

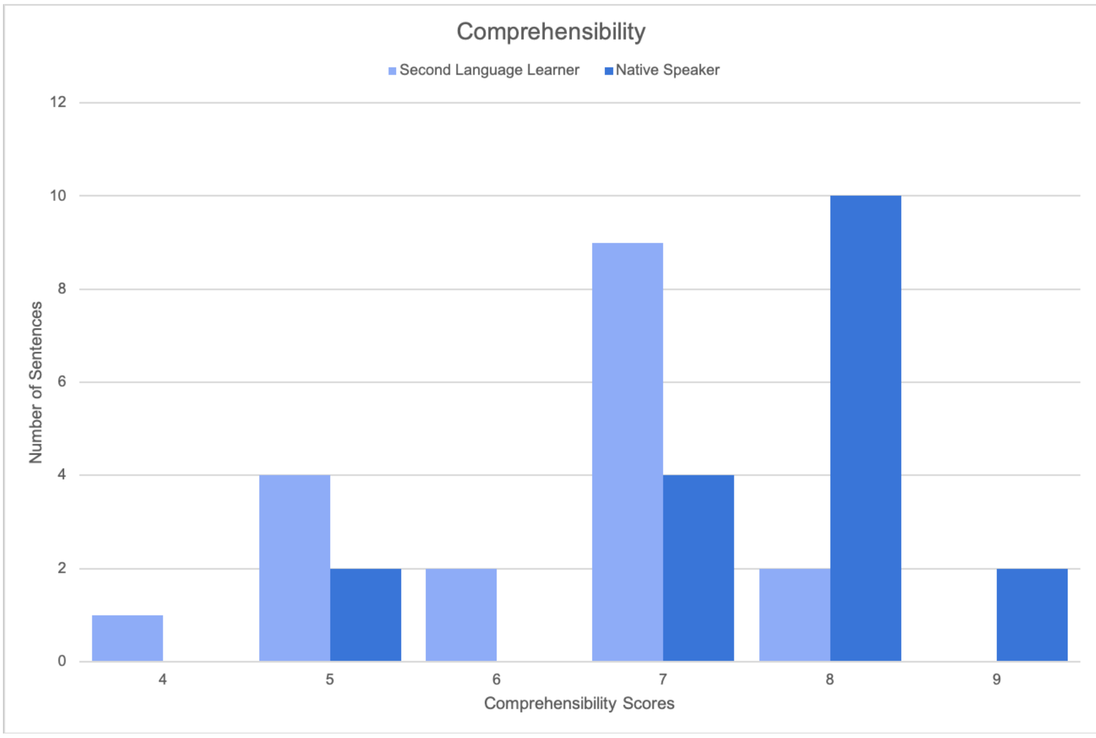


Figure G.1: Comprehensibility (Average Score of 6.97)

[1]Scores that were never chosen by the evaluators are not shown.

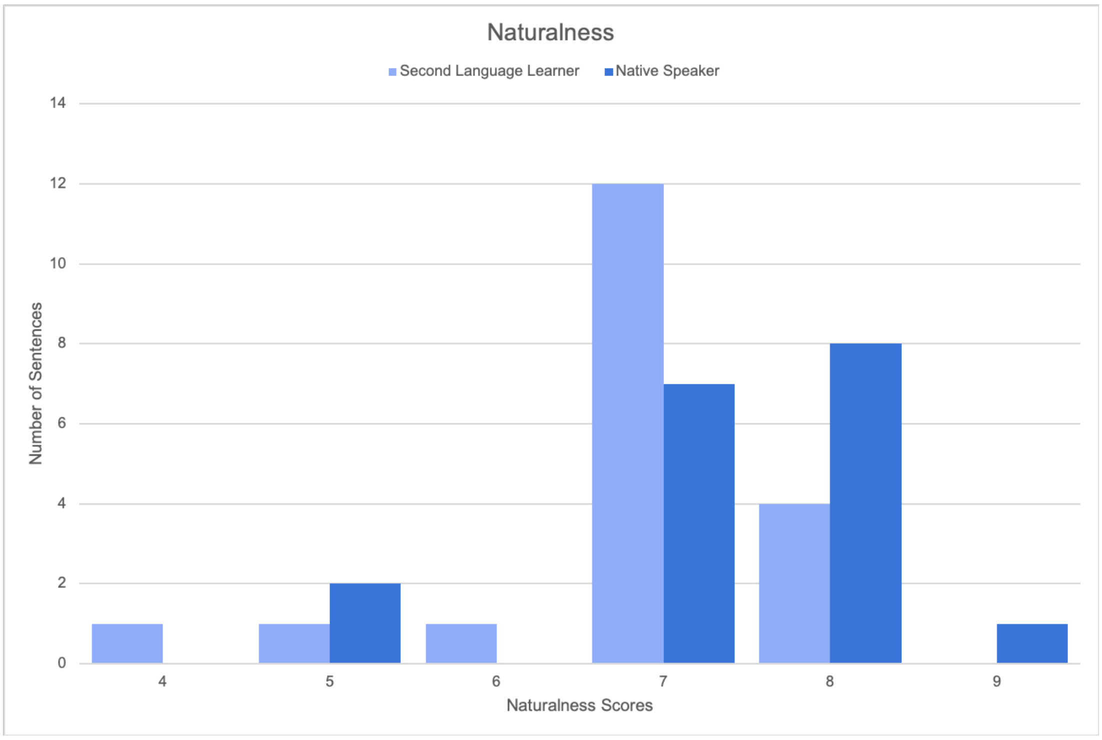


Figure G.2: Naturalness (Average Score of 7.31)