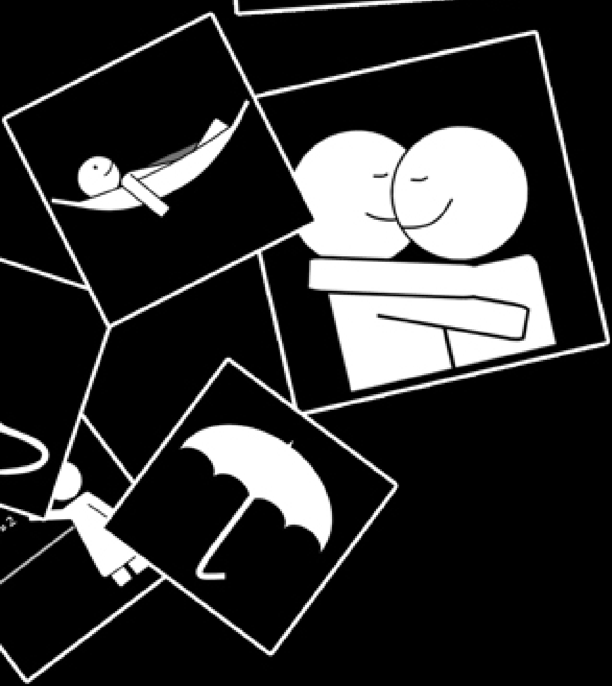


WORDS DIVIDE PICTOGRAPHS UNITE

Pictograph Communication Technologies for People with an Intellectual Disability

Leen Sevens

Words Divide, Pictographs Unite

Pictograph Communication Technologies for People with an Intellectual Disability

Cover design: Leen Sevens
Pictographs on cover: Sclera (http://www.sclera.be/)

KU Leuven
Faculteit Letteren
Onderzoekseenheid Taalkunde
Onderzoeksgroep Computationele en
Formele Taalkunde
Centrum voor Computerlinguïstiek

Words Divide, Pictographs Unite

Pictograph Communication Technologies for People with an Intellectual Disability

Leen Sevens

Proefschrift ingediend tot het
verkrijgen van de graad van
Doctor in de Taalkunde

Leuven, 2018

Promotiecommissie

Promotor:	Prof. Dr. Frank Van Eynde
Co-promotor:	Dr. Vincent Vandeghinste
Leden:	Prof. Dr. Walter Daelemans
	Prof. Dr. Ilse Noens
	Prof. Dr. Horacio Saggion
	Prof. Dr. Frieda Steurs

To my family.

For most people, technology makes things easier.
For people with disabilities, technology makes things possible.
— Mary Pat Radabaugh (2014)

Acknowledgements

The present dissertation, resulting of four years of research, would not have been possible without the help, encouragements, and support of many people.

First and foremost, I would like to thank my supervisor Prof. Dr. Frank Van Eynde for offering me the possibility to freely conduct my doctoral research in such a stimulating environment. His feedback, insight, and encouragements were never missing, and his door was always open for me. He inspired me to stick to the highest scientific standards and I am determined to continue doing the same throughout my career. I also want to express my gratitude to my co-supervisor, Dr. Vincent Vandeghinste. He often believed more in my abilities than I did myself, and he convinced me to aim for highly ranked publications, enter scientific competitions and events, and experiment with state-of-the-art techniques. I am forever grateful for his valuable feedback and creative ideas, and for providing the basis for my dissertation, without which this research would not have been possible. Frank and Vincent, thank you so much for introducing me to the magical world of computational linguistics, inspiring me to study Artificial Intelligence, giving me the opportunity to do my internship at the Centre for Computational Linguistics, and allowing me to stay. I would not be where I am as a researcher today without your expertise, kindness, and support.

A big thank you goes out to Liesbeth Augustinus, with whom I have shared an office since the very first day of my internship at the Centre for Computational Linguistics. Academia can be a scary place for a sensitive human being like myself, but Liesbeth was always there for me to make me feel at ease. Thank you for bringing me coffee and tea during my first days at the office, dragging me outside for lunch, listening to my endless rants, helping me out with my doctoral paperwork - seriously, you need a PhD for that -, and above all, thank you for being such a great friend. You were the best office mate I could wish for.

I am forever grateful to Ineke Schuurman for ensuring that I became a full-fledged and worthy member of the Able to Include consortium, and for teaching me the ways of project proposal writing. Thank you for travelling with me to faraway places, involving me in the organisation of a social inclusion workshop in the Land of the Rising Sun, enlightening us with your Dutch expressions, and taking the time to correct my least favourite part of this dissertation: the bibliography.

I want to express my gratitude to Tom Vanallemeersch, who stole my desk quickly after my internship ended, but made up for that crime by entertaining me with random movie facts, taking me to the Thai House on Fridays (but never on Mondays!), adding even more *memes* to my linguistic jokes wall, and showing genuine interest in my work. Tom, thank you for sharing your expertise with me, helping me think out of the box and bringing unusual solutions to the table, and inviting me to tackle a shared task on the normalisation of medieval Dutch text together.

I am also grateful to the external members of the Centre for Computational Linguistics, Peter Dirix and Bert de Bruijn, for the discussions we had on linguistic and technical issues. Peter, thank you for your expertise, brain teasers, and above all, your friendship. Bert, I have to admit that I would sometimes freak out when our machines were not really behaving themselves, or when I was unable to install some obscure new package or tool, but you were always just one call or email away. Thank you so much for your help.

Dear (ex-)interns at the Centre for Computational Linguistics, I am very sorry for starting this tradition where interns need to host a party when they leave our team. But I also want to take this opportunity to thank you for your hard work, and for helping me grow as a mentor myself.

In short: Dear colleagues at the Centre for Computational Linguistics, thank you for all the great moments we shared - our weekly trips to Alma (*Let's hope they serve pita!*) and Le Mustache (*Let's hope they don't give us Koetjesrepen!*), the frequent office parties that we hosted to celebrate virtually anything and everything, our linguistic adventures abroad, and so on. We were a very small, yet tightly-knit team, and I am proud to have been part of it.

My thanks go equally out to the members of my advisory committee and my jury for providing me with their precious advice on the topics of natural language processing, inclusion, and disability studies. I would like to express my eternal gratitude to Prof. Dr. Walter Daelemans, Prof. Dr. Ilse Noens, Prof. Dr. Horacio Saggion, and Prof. Dr. Frieda Steurs for reading and approving my work.

My gratitude goes out to the Agentschap Innoveren & Ondernemen (previously IWT) for funding my doctoral research, and to the Steunfonds Marguerite-Marie

Delacroix, who made this research line possible in the first place. Above all, I would like to thank Jan Dekelver and the WAI-NOT team for entrusting the Centre for Computational Linguistics with the task of developing pictograph translation technologies, implementing our solutions on the WAI-NOT website, recruiting participants for our user surveys, and providing us with valuable user data and feedback. Tony Pijls, Nancy Van Belleghem, Marthe Nelissen, and Alexander Van Paemel, keep up the great work. It has been an absolute pleasure working with you.

I would further like to express my gratitude to the members of the Able to Include project (funded by the European Commission - Competitiveness and Innovation Framework Programme), in which I was able to carry out the first stages of my research. My sincerest thanks go out to the Universitat Pompeu Fabra (ES) and the Microelectronics Applications Centre (IE) for collaborating with me in creating accessible language technologies and apps for people with an intellectual disability. Thank you, Fundación Prodis (ES), Building Bridges Training (UK), and K-point at Thomas More (BE) for allowing users to experiment with our services and for providing me with valuable feedback. I would also like to thank Inclusion Europe (BE) for spreading the word about our project. We were a great team.

I also greatly appreciate the interest and advice of our research group, OG ComForT, and the encouragement of the staff of the Faculty of Arts and the Conference & Events Office at KU Leuven. You have been exceptionally supportive of my work, cheered me on in national and international competitions, communicated about the progress of my research with the outside world, and involved me in all sorts of exciting science communication events. Thanks to all of you, I have discovered my passion for creative science communication, and I have learned that being myself does not decrease the value of my research. Thank you so much for giving me the opportunity to share these findings with my peers.

During my research, I have been in touch with many researchers, disability organisations, schools, day centres, and families. To avoid the risk of forgetting someone, I would like to express my full-hearted thanks to everyone who contributed to my research project in one way or another. Thank you so much, all of you, for your valuable feedback, your genuine interest, your fruitful discussions, your critical remarks, and your constant support. My gratitude goes equally out to the Sclera and Beta organisations, who created new pictographs for us to use in the technologies. And of course, I would like to thank the end users for helping me understand their struggles and needs, and for wholeheartedly believing in the potential of the developed solutions.

Last, but certainly not least, I would like to thank my family and friends for their endless support. Their (virtual) pats on the back, words of encouragement, and irrelevant distractions helped me stay sane and happy throughout my research and writing process. Thank you, bomma, uncle, aunt, cousins, and the family of my fiancé, for your encouragements and moral support. I am also forever grateful to my lunch buddies, my friends from high school and university, my online gaming mates, the Cheesybread squad, my fellow Pokémon trainers, and all my other friends for cheering me on and always putting a smile on my face.

An Brouns and Filip Sevens, mama and papa, I am grateful for your immense love and support throughout my life in achieving my dreams. Thank you for always believing in me, and helping me be the best person I can be. Mama, thank you for all those times you brought home-made (cup)cakes and cookies to my office, for treating me to cocktails and dinner every week, for being my loudest supporter - often quite literally! -, and for making me the joyful person I am today. Papa, thank you for passing on the linguistic gene to me, for showing such an enthusiastic interest in my work, and for giving me the freedom to pursue my dreams.

Finally, I want to thank Niels Berghen, my fiancé, for standing beside me throughout my academic career and writing this dissertation. Niels, thank you for giving me strength and encouraging me to reach my goals. I could not have done it without your unconditional love and continued support. Thank you for all those times you bought me flowers and cooked my favourite meals to celebrate my return from a conference trip, for exploring the world with me, for raising a toast to all the small victories, and for bringing so much colour into my life. If you were a library book, I would never bring you back.

Contents

Preface

Words divide, pictographs unite. The title of the present dissertation echoes a quote from the Austrian philosopher and sociologist Otto Neurath (1882-1945), who claimed the superiority of pictorial languages over verbal languages by means of a catchphrase that would later become a motto for graphic designers from all over the world: “words divide, pictures unite”. With a primary aim to communicate empirical observations, Neurath believed his method of pictorial statistics to be an effective tool to transmit rational information across barriers of class, culture, and education (Nikolow 2011:86). Even today, in the line of Neurath’s work, students and researchers use infographics and smart data visualisations in an attempt to reach a wider audience for their scientific stories.

We are convinced that, in today’s society, the applicability of Neurath’s quote is much wider than originally intended. Even more, when considering people with an intellectual disability (ID), these four words do not only capture the essence of what it feels like for them to venture on the Internet (“words divide”), they also inherently provide us with a solution to the problem (“pictures unite”).

As a literate individual who is living in the era of technology, it can be hard to imagine what it feels like to be e-excluded from today’s digital society. Let us give you a sense of the scope of the issue and the impact of the developed solutions by splitting the quote into two parts.

Words Divide

Imagine this. You are stuck on a planet in a parallel universe where everyone speaks an alien language. You are trying to fit in, but communication is difficult, as there are no translation tools available. You are missing out on events and parties, you cannot stay in touch with your friend who recently moved to the other side of the planet, and you find it difficult to keep up with the latest trends. You cannot help but feeling lonely and left out.

For many people with ID, this situation is far too real. By that, we do not mean that they are not able to use the language of extraterrestrial beings, but that social media and email are not cognitively accessible for them due to severe reading and/or writing difficulties in their own native language.

Never in the history of mankind have we produced more text than at this present moment - especially when considering online platforms.[1] Being able to read and write is an important way of taking part in society. However, we tend to forget that, even in educated communities, not all people can read or write, and that there exist several degrees of literacy. Written text creates a barrier between people with and without ID. As a result, they are e-excluded.

Pictures/Pictographs Unite

Luckily, picture-based communication is no science-fiction.

Schools and day centres for people with ID already use specialised picture systems, called *pictographs*,[2] that depict everyday activities and objects, to enable accessible written communication between children or adults with ID and their caregivers. These pictographs appear in print, on menus, schedules, and calendars, and represent all sorts of information that may be difficult to read or express in written text. They are easy to recognise and memorise, making them also a useful tool for people with an autism spectrum disorder. However, while many of the end users are "fluent" pictograph users, they do not have access to these pictographs in online environments, where they are still confronted with large amounts of unintelligible text.

In this dissertation, we transfer these pictographs to social media and email, by developing translation tools that help people with ID stay in touch with their family and their friends. The focus in this work lies on Dutch and two pictograph sets that are commonly used in Flanders,[3] but we will show that the language technologies can be extended to other natural languages and pictograph languages, as the baseline systems are designed to be as language-independent as possible.

[1]In 2014, there were 2.4 billion Internet users. That number grew to a total of 3.8 billion Internet users by April, 2017 - a 42% increase in people using the Internet in just three years. Data trackers reveal a continuous increase in the number of tweets and Facebook posts shared per minute, and in the number of emails and text messages sent per day. For instance, between 2013 and 2017, the number of tweets shared per minute had increased by 58%. That corresponds to more than 455,000 tweets per minute. See also: https://blog.microfocus.com/how-much-data-is-created-on-the-internet-each-day/

[2]Also called *pictograms*.

[3]The Dutch-speaking part of Belgium.

Structure of this Dissertation

This dissertation is organised into eleven chapters, which are spread across four parts.

In **Part I**, the *Prologue*, we first provide an overview of the semantic properties, communicative functions, and formal characteristics of pictographs, with special attention to pictographs that are designed for people with ID (**Chapter 1**). There exist as many definitions for the concept known as *pictograph* as there are scholars to define it. Our challenge lies in finding a unified definition. In the next chapter, we provide some background information (**Chapter 2**). We discuss the origins of the pictograph translation project, define the target users, and explain how the pictographs that are used in the technologies are connected to lexical-semantic databases called WordNets.

Part II is dedicated to the *Text-to-Pictograph* translation engine, a language technology that automatically translates Dutch natural language text into pictographs for people with reading difficulties. We first discuss the shortcomings of the baseline translation system (**Chapter 3**), and then proceed to the description and evaluation of three major improvements: We create a spelling correction tool for people with ID (**Chapter 4**), we develop a syntactic simplification tool and a temporality detection module (**Chapter 5**), and we implement a word sense disambiguation tool for improved semantic analysis (**Chapter 6**). The added value of each one of these components is measured by a combination of automated metrics, manual evaluations, and, where possible, user studies. Finally, we evaluate the improved Text-to-Pictograph translation pipeline (**Chapter 7**).

Part III is concerned with the other direction. The *Pictograph-to-Text* translation tool provides help in constructing Dutch textual messages by allowing a user to input a series of pictographs, and then translates these messages into natural language text. The challenge in Pictograph-to-Text translation is twofold. The first task involves the development of an accessible interface that allows people with ID to find the pictographs of their choice (**Chapter 8**). The second task concerns the actual development of the Pictograph-to-Text translation engine (**Chapter 9**). We discuss a variety of approaches, including language modelling and machine translation techniques, toward the generation of rich natural language text from underspecified pictograph input.

In, **Part IV**, *Extensibility and Comparison*, we first demonstrate the extensibility of the translation engines toward other natural languages and other pictograph languages (**Chapter 10**). As a case study, we present the development of the English and Spanish baseline translation systems. Finally, we provide an overview of other pictograph-based technologies for local and remote communication (**Chapter 11**). A large part of this chapter is dedicated to the solutions that have been developed for people with ID, and their practical and/or technical limitations, as compared to our own translation technologies.

These chapters are followed by a conclusion and seven appendices containing supplementary information and data. We enlist some of the commonly used abbreviations in Appendix A. The development set and test set that are used in most of the experiments throughout this dissertation are shown in Appendix B, as well as their manually corrected forms.[4] Appendix C contains the (translated) Klare Taal checklist. The pseudocode for the syntactic simplification module is provided in Appendix D, while Appendix E describes the participants of the user survey. The hierarchical structure of the static pictograph interface is presented in Appendix F. Appendix G, finally, gives an overview of the object-oriented framework.

[4]All other data used throughout this dissertation are available on request.

Part I

Prologue

Chapter 1

Status Quæstionis: Pictographs

What is in a name? Perhaps the best and most straightforward way to define the concept known as *pictograph* is by having a look at the etymology of the word in the first place: *pictus* (Latin) and *graphé* (Greek), *painted writing*, "written communication by means of images". Take a glance around you, and you might notice that the power button of your computer, its microphone and headphone jacks, the label of your water bottle, the frost protection mode of your heater, and all the applications on your computer's desktop are provided with small, recognisable, graphic representations that support or replace written textual instructions or labels. Pictographs are found everywhere around us. Nevertheless, the practice of using hand-drawn images as a way to convey meaning is certainly not an invention of our time. In fact, the first known written languages were *pictographic* (Chen 2004:3).

One of the earliest cases of storytelling through visual representation are the prehistoric cave paintings, which were made from powdered minerals, charcoal, or other substances. The Lascaux Caves in France, for example, contain over 2,000 paintings dating from 15,000 to 10,000 B.C. and reveal what wild animals lived during the Paleolithic Era, among other things. In a more recent past, the Egyptians (3100 B.C. to 400) and Maya (300 B.C. to 1697) used sophisticated hieroglyphic writing systems for the recording of history, the administration of social systems, and creative expression (Innocent 2001:255). While these picture-based systems have morphed into alphabets over time, which depend more heavily on fixed conventions than on similarity for their representational power (Tanimoto 1997:2),[1] pictographs have not yet lost their

[1]Around 1400 B.C., the Phoenicians created an *abjad*, a writing system where each symbol or glyph stands for a consonant, with signs derived from the Egyptian hieroglyphs. In the Phoenician alphabet, signs represent the initial *sound* of the name of the depicted object, instead of the *meaning*

effectiveness as carriers of meaning. Think of traffic signs, signs for telling people that dogs should be kept on a leash, or signs in buildings that direct visitors to the elevators, the meeting rooms, and the emergency exits.

This chapter provides an overview of the **meaning**, **function**, and **form** of pictographs in the Modern Era (1945 - present), with special attention to pictographs that are designed for people with an intellectual disability (ID). In the light of Peirce's sign theory (Peirce 1908), section 1.1 discusses the subtle differences between icons, indices, and symbols. As will be shown in section 1.2, the use of pictographs as a means of communication has clear advantages for people with and without ID, although the goal of creating a universal and culture-transcending pictograph communication system will always remain difficult, if not impossible, to achieve. Section 1.3 sketches some of the commonly accepted criteria for pictograph design and zooms in on the formal characteristics of five pictograph sets: Semantography, Visual Inter Lingua, emoji, Sclera, and Beta. Based on the observations that were made in the previous sections, we propose a unified definition in section 1.4.

1.1 Meaning: The Semantic Properties of Pictographs

The semantic relation between pictographs (i.e., the visual representations, the images as such) and their intended referents is complex in nature and can be defined in numerous ways. As we shall argue in section 1.1.1, we opt to define this relation in terms of Peirce's sign theory and the traditional division of signs into icons, indices, and symbols. Peirce's sign theory accounts for *any* type of signs, not just linguistic signs - which are the focal point of Saussure's sign theory (Saussure 1916) -, even though pictographs, the way we know them today, did not yet exist at the time. More recent accounts of sign studies have proposed all sorts of definitions for the wide range of representation strategies that may be used by an interpreter to associate a pictograph with its referent. However, as we will discuss in section 1.1.2, many of these descriptions are incomplete, and the proliferation of terms that intrinsically re-

of that object. For instance, the Egyptian picture of a house or a tent's floor plan, typically depicting a house or a tent (and a number of related meanings, such as "family"), was slightly modified and given the Phoenician name *beth* 'house', representing the sound "b". Note that, apart from word-signs (i.e., pictures of objects used as the words for those objects), Egyptian hieroglyphic signs could also be syllabic signs (i.e., signs which stood for syllables). In Greece, around 900 B.C., the Phoenician script was further modified and vowels were added, giving rise to the first true alphabet.

fer to the same thing has prevented scholars from putting forward a unified definition. In section 1.1.3, we apply Peirce's traditionally defined icon-index-symbol triad to the pictograph case. Section 1.1.4 concludes.

1.1.1 Traditional Sign Theories

Charles Sanders Peirce's *sign theory* or *semiotics*, the study of signs, constitutes one of the earliest accounts of signification and representation. In a letter to Lady Welby-Gregory in 1908, Peirce proposes the following definition:

> "I define a Sign as anything which is so determined by something else, called its Object, and so determines an effect upon a person, which effect I call its Interpretant, that the latter is thereby mediately determined by the former." (Peirce 1908)

In other words, Peirce distinguishes between the *signifier* or *representamen*, such as a written word or an image, the *object*, which is signified by the signifier, and the *interpretant* or the *understanding* of the relation between the signifier and the object. The key role of the interpretant is one of the most innovative features of Peirce's sign theory. It is abstract in nature and it does not exist in human perception (Yakin & Totu 2014:7). An example of a Peircean sign is given in Figure 1.1. Note that the object is not a word or a phrase, but a real thing or a concept, to which the signifiers refer.

Peirce (1867) states that signs can be interpreted or understood in three possible ways, hereby introducing the original tripartition between *icons*, *indices*, and *symbols*:

- **Icons** or **likenesses** are established via shared qualities. Visual icons physically resemble the object or entity they stand for. The photograph or drawing of an elevator, for example, is an icon of an elevator. Auditory icons, such as onomatopoeia, phonetically resemble the sound of the object that they describe.
- **Indices** are generated when there is a correspondence or correlation between the signifier and the object, such as a causal relation. In this case, the relation is said to be mediated. Dark clouds in the sky may predict rain in the near future. Indexical expressions, like the words *here* or *there*, indicate the relative distance of people and things within a given context.

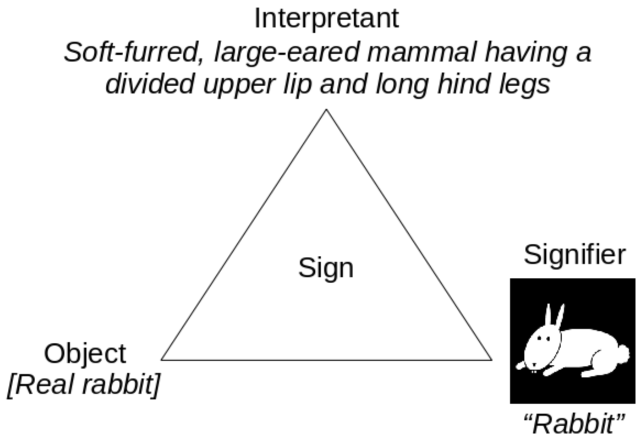


Figure 1.1: Peircean signs, as illustrated by two signifiers: a pictograph (i.e., the image as such) and an English word.

- **Symbols** are formed without depending on any correspondence in space and time between the signifier and the object. Their connection is general and purely conventional or arbitrary. Natural languages are permeated by symbols. The words *owl* (English), *hibou* (French), and *uil* (Dutch), for instance, all share the same referential object.

Whether a sign is interpreted as an icon, an index, or a symbol is up to the interpreter and his/her previous knowledge, experiences, or expectations. For example, if the image of a dark cloud would refer to a dark cloud as such, the relation between the signifier and the object would be iconic, instead of indexical.

Saussure (1916) gives us a similar account of signs. Yet, there are a number of differences with Peirce's sign theory. For Saussure, signs are entities that are composed of two elements: a *signified* (the mental concept) and a *signifier* (the sound pattern).[2] An example of a Saussurean sign is given in Figure 1.2. It is important to note that Saussure's sign theory is largely restricted to the field of natural languages. As opposed to Peirce, whose signs extend into the physical world, Saussure considers signs, primarily, as an aspect of word construction. Saussure's signifier and signified are inseparable; they do not exist without each other, creating meaning simultaneously,

[2]Saussure's *signified* is often compared to Peirce's *object*, whereas Saussure's *signifier* is often compared to Peirce's *signifier*, although Peirce's *signifier* is a broader category than Saussure's *signifier*.

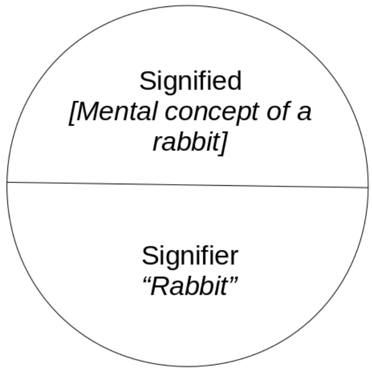


Figure 1.2: Saussurean signs, as illustrated by the signifier “rabbit”, an English word. The signified is the mental concept of a rabbit.

together (Yakin & Totu 2014:7).[3] An important difference with Peirce’s sign theory is that Peirce introduced a third element that is necessary for the signification to occur, namely, the interpretant. In other words, the signifier only accesses its signified through interpretation.

Peirce’s theory can, in theory, account for all kinds of signs, including pictures, or smoke as an indicator of fire, and not just linguistic signs. This observation makes Peirce’s sign theory more suitable for our analysis of pictographs as signs.

In the following section, we provide an overview of contemporary sign studies on the topic of pictographs, and discuss the analogies and (often subtle) differences between these theories and Peirce’s traditionally defined icon-index-symbol tripartition, while highlighting the incompleteness of the proposed analyses.

1.1.2 Contemporary Analyses of Pictographs as Signs

Although definitions vary considerably, the relationship between pictographs and their referents is nearly always described in terms of at least one of Peirce’s sign types. It is important to note that different researchers often use different terminology,

[3] The fundamental principle of the arbitrary nature of the linguistic sign does not prevent Saussure from distinguishing between what is intrinsically arbitrary (i.e., unmotivated) and what is only relatively arbitrary (i.e., motivated). According to Saussure, a sign may be motivated to a certain extent. For instance, signifiers must constitute well-formed combinations of sounds. Furthermore, compound nouns, such as *haircut*, are not fully arbitrary, since they are a meaningful combination of two existing signs.

sometimes to describe “what, in essence, is the same thing” (Wang et al. 2007:203). However, most scholars make no explicit mention of the traditional icon-index-symbol tripartition. We present an overview.

1.1.2.1 The Icon Definition

A considerable amount of contemporary sign studies place pictographs on the same level as Peircean icons. The following definitions are incomplete, suggesting that a pictograph must always physically resemble the object or entity they stand for. As we will argue in section 1.1.3, this is not always the case.

The graphical images that are used in the Visual Inter Lingua (VIL) system created by Leemans (2001:39) are referred to as *icons*, as “they rely initially on recall of a previous visual experience on the part of the user”. Similarly, Tatomir (2003:10) proposes a pictograph system for visual communication, the elements of which she calls *icons*, defined as “a mode in which the signifier is perceived as resembling or imitating the signified, being similar in possessing some of its qualities”.

Remarkably, in Modern English, the “small graphic representations” (Maiti et al. 2011:173) that are displayed on computer desktops or mobile devices are also referred to as *icons* or, interchangeably, *pictographs*. We should note that these *digital* icons do not necessarily correspond to *Peircean* icons. Rather, they are signifiers that can be interpreted in iconic, indexical, or symbolic ways. For instance, the *trash can* pictograph on a desktop indirectly refers to the act of deleting data; it does not refer to a real trash can. The modern use of the term *icon* as a designation for any digital, graphical sign suggests that its original meaning has become more general with the rise of technological advances. In the remainder of this dissertation, we will reserve the term *icon* strictly for its Peircean sense.

1.1.2.2 The Icon Versus Symbol Definition

A number of contemporary sign studies recognise the differences between iconic and symbolic pictographs, but they leave no room for indexical relations, such as metaphors or facial expressions as an indicator of human emotion.

Marcus (2003:38) defines pictographs as “icons, or sometimes symbols”, that have clear visual similarities with some object. Following the Peircean tradition, Marcus identifies *icons* as self-evident, natural, or realistic signs, like a photograph of a person, while *symbols* are only meaningful by convention, like the letters of this sentence or a national flag. Cho et al. (2008:65), as well, recognise the traditional division between icons and symbols, claiming that a *pictograph* should not be confused with a *pictorial symbol*. They define pictographs as *icons* with a clear pictorial similarity with some object, whereas pictorial symbols are not universally interpretable. The depiction of an arrow, for instance, could suggest directionality to some people. Other people might recognise it as a sign for war or bad luck. Therefore, the depiction of the arrow would be a pictorial symbol.

It is worth mentioning that the International Organisation of Standards (ISO), an international standard-setting body, refers to *graphical symbols* as "a means of visual pictorial communication, intended to give information to the general public" (ISO 1984). According to ISO's definition, traffic signs, signs in airports, and other *pictographs* are examples of international *symbols* that can be understood across many languages and cultures. In addition to this, and building on ISO's description of *symbols*, Cowgill & Bolek (2003:7) propose a distinction between *image-related* or *pictorial symbols*, where the image resembles the referent, and *concept-related* or *arbitrary symbols*, where the image has no relation to the referent. Terminology-wise, it can be inferred that ISO's *symbol* does not correspond to the Peircean *symbol* in a one-on-one fashion.

1.1.2.3 Fine-Grained Taxonomies

A number of contemporary pictograph taxonomies are similar to or go well beyond the Peircean tripartition between icons, indices, and symbols. While the following taxonomies resemble the traditional icon-index-symbol tripartition, different terminology is often used to refer to the same concepts.

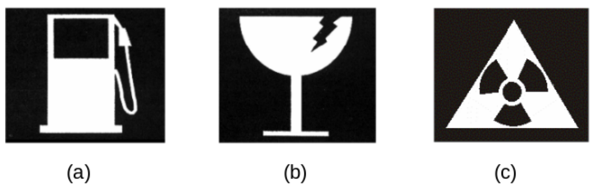


Figure 1.3: Pictograph taxonomy according to Lodding (1983:16): (a) representational (*petrol pump*), (b) abstract (*fragility*), and (c) arbitrary (*radiation*). Pictographs taken from Wang et al. (2007:204).

Lodding (1983:16) identifies three types of pictographs: *representational*, *abstract*, and *arbitrary* pictographs (see Figure 1.3). Representational pictographs serve as an example for a general class of objects, displaying "many of the features that an instance of this class of objects would have", such as the image of a (prototypical) petrol pump to represent any petrol pump *(a)*. Abstract pictographs present a concept to a viewer that is associated with the physical object, rather than features of the object itself, like the image of a broken glass to represent fragility *(b)*. Finally, arbitrary pictographs are invented and assigned a meaning that cannot easily be discovered from the context, such as the sign for radiation *(c)*. Using different terminology for the same concepts, Purchase (1998) distinguishes between *concrete* pictographs, *abstract* pictographs, and *symbolic* pictographs, while Gaver (1986:170) speaks of *nomic*, *metaphorical*, and *symbolic mappings*. Note that Gaver's analysis goes further than most other taxonomies in that it introduces two subtypes of metaphorical mappings: *Structure mappings* exploit the similarities between the structures of two things, such as the use of a tree to represent genealogy, while *metonymic mappings* use a feature to indicate a whole, such as the use of a horseshoe to represent a horse.

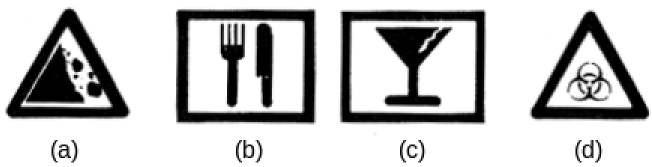


Figure 1.4: Pictograph taxonomy according to Rogers (1989): (a) resemblance (*rocks falling*), (b) exemplar (*restaurant*), (c) symbolic (*fragility*), and (d) arbitrary (*biohazard*). Pictographs taken from Rogers (1989:110).

Rogers (1989:110) recognises no less than four pictograph types: *resemblance, exemplar, symbolic*, and *arbitrary* (see Figure 1.4). Lidwell et al. (2003) propose a nearly identical taxonomy, merely replacing the term *resemblance* with the term *similar*. Resemblance or similar pictographs depict their referent using an analogous image, like the road sign that warns drivers for falling rocks *(a)*. Exemplar pictographs show only the most central attributes of an object, such as a knife and fork denoting the presence of a restaurant *(b)*. Symbols convey an underlying referent that is at a higher level of abstraction than the symbol itself. An example of this would be a broken wine glass to imply fragility *(c)*. Finally, arbitrary pictographs have no relation to their referent, such as the biohazard sign *(d)*. In our view, Rogers' resemblance and arbitrary (i.e., Lidwell et al.'s similar and arbitrary) pictographs correspond to Peircean icons and symbols, respectively, while the exemplar and symbolic pictographs are, rather, two subtypes of Peircean indices; the relation between these images and their referents is indirect or mediated.

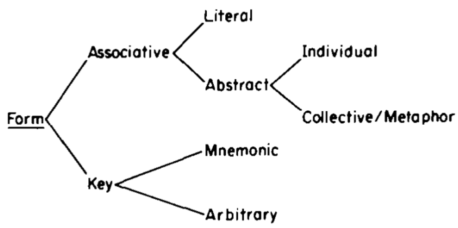


Figure 1.5: Pictograph taxonomy according to Gittins (1986). Taken from Gittins (1986:529).

Gittins (1986:529-530) considers the use of pictographs within the domain of human-computer interaction - an interesting perspective, given that we will be transferring pictographs to the digital world. Gittins defines pictographs as either "manifesting a characteristic of the underlying object", in which case he calls them *associative* pictographs, or "serving as a *cognitive key* to it" (see Figure 1.5). Associative pictographs involve an inference process. While this definition is relevant for the so-called *abstract* pictographs (such as the ones that relate to the metaphorical desktop, like folders, files, trash cans, calendars, and so on), it is remarkable that Gittins considers *literal* pictographs (which are left undefined, but which are likely to be some sort of equivalent to Peircean icons) as a subtype of associative pictographs, given that iconic pictographs typically do not require an inference process. Cognitive keys, on the other hand, can either be *mnemonic* or *arbitrary*. Mnemonic pictographs remind the user of the commonly accepted name of an object. For instance, the process of executing a command in an operating system might be represented by the image of a guillotine. We should observe that mnemonic pictographs are heavily language-dependent and, generally, tend to be avoided in graphic design. Real-world examples of mnemonic pictographs are, in fact, very difficult to find.[4] Finally, arbitrary pic-

[4]As noted by Leemans (2001:138), images that rely on inside jokes, figures of speech, slang, or other terminology that is well-known only within a particular subculture, will be intelligible only to members of that group. An example of this would be the image of an ant, trapped in a jar, representing the process of *debugging*.

tographs have a form of arbitrary design, from which it is not possible to infer the nature of the object, such as the international pictograph for radioactive material.

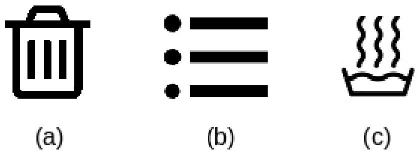


Figure 1.6: Pictograph taxonomy according to Blattner et al. (1989): (a) representational (*trash can/delete*), (b) abstract (*list*), and (c) semi-abstract (*steam*). Pictographs taken from Blattner et al. (1989:16-18).

A number of taxonomies hardly resemble the traditional icon-index-symbol tripartition, as they do not use semantics as a distinguishing criterion. Blattner et al. (1989:16-18) propose the distinction between *representational*, *abstract*, and *semi-abstract* pictographs (see Figure 1.6), using **formal** characteristics as a primary indicator of pictograph type. Representational pictographs are simple pictures of familiar objects or operations, such as a fork and a knife *(a)*.[5] Abstract pictographs are composed of geometric shapes and other non-recognisable figures *(b)*, while semi-abstract pictographs are composed of both representational and abstract images *(c)*.

Ma et al. (2015:72), finally, describe pictographs within the context of computer interfaces, using **function** as a way to distinguish between *action* pictographs and *knowledge* pictographs. Action pictographs guide users to perform specific operations on an interface. An example of this would be the three letters on the toolbar of Microsoft Word, allowing a user to create bold, italic, or underlined text. Differing from action pictographs, knowledge pictographs are those passing on some information to users, but not for action purposes. For instance, an hourglass with different percentages of sand represents the different stages of an installation procedure.

This overview shows that the semantic properties of pictographs are difficult to define, with as many taxonomies as there are scholars to define them. More specifically, the terminology used to describe pictographs has grown to become remarkably complex and confusing over the years. In the next section, we will redefine the semantic properties of pictographs in terms of the original Peircean tripartition, with an application to pictographs that are designed for people with ID.

[5]From a Peircean point of view, a fork and a knife would indicate the presence of a restaurant, in which case the relation is said to be mediated.

1.1.3 Analysis of the Semantic Properties of Pictographs with an Application to Pictographs Designed for People with ID

We present a fine-grained analysis of the potential relations between pictographs and their referents, as expressed in terms of the Peircean triad. Sclera examples are provided (see section 1.3.2.3.1). The Sclera set is designed for people with ID, and it is one of the two pictograph languages that are used in the Text-to-Pictograph and Pictograph-to-Text translation systems that are described in the following chapters.

1.1.3.1 Iconic Relations

There can be an immediate or *iconic* connection between the pictograph and its referent through visual similarity. In airports, the pictograph of an elevator represents the elevator, the pictograph of a piece of luggage on a conveyor belt represents the conveyor belt, and the pictograph of a bus represents any bus. Many pictographs that are used in communication systems for people with ID are iconic (see section 1.2.1.3), with most of the images referring to people, animals, and everyday objects (see Figure 1.7).

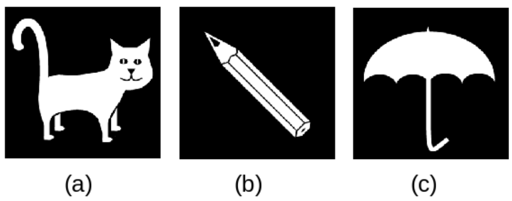


Figure 1.7: Sclera examples of iconic relations: (a) *cat*, (b) *pencil*, and (c) *umbrella*.

Drawings are an effective way to represent static objects (Tijus et al. 2007:23). However, as noted by Leemans (2001:37) and Nakamura & Zeng-Treitler (2012:543), only physical objects with an identifiable shape are eligible to be represented through visual similarity. Conceptual entities, such as *envy*, and events, such as *destruction*, cannot be depicted reliably. Tijus et al. (2007:23) consider the difficulty of using drawings to represent categories of objects, especially those that group objects of different shapes, such as *fruit* or *animal*. Finally, Zhou (2014:164) acknowledges that some subjects are not suitable to be expressed with images at all. For instance, logical and philosophical theories can simply not be expressed through iconic means. This is where indices and symbols come into play.

1.1.3.2 Indexical Relations

When the relation between a pictograph and its intended referent is mediated, the connection is said to be indirect or *indexical*. In this case, the pictograph still maintains a relation with its referent, although that relation is less explicit (Purchase 1998:9). Representation through semantic association can be realised in various ways. The following classification is proposed by Nakamura & Zeng-Treitler (2012:545-546). Sclera examples are shown in Figure 1.8.

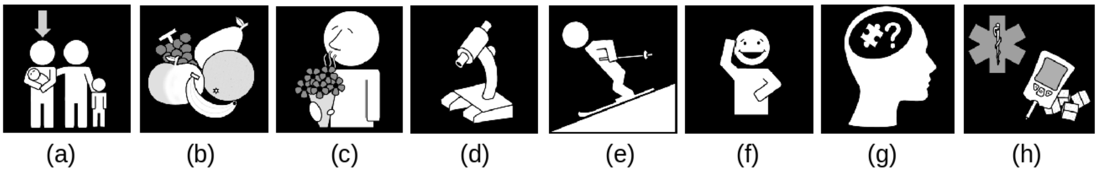


Figure 1.8: Sclera examples of indexical relations: (a) *mother*, (b) *fruit*, (c) *to smell*, (d) *laboratory*, (e) *skiing*, (f) *happy*, (g) *autism*, and (h) *diabetes*.

(a) **Comparison** or contrast is used to represent a main concept in relation to other, accessory elements. For instance, arrows make a *mother* stand out in a *family* pictograph. This strategy is also commonly used for depicting adjectives.

(b) **Exemplification** or the use of multiple examples to represent a category allows for depicting collective concepts, such as *fruit* or *animal*.

(c) **Semantic narrowing** is similar to exemplification, but only uses one example. For example, verbs that require an object are represented with one specific object. A flower can be depicted to clarify the verb *to smell*.

(d) **Physical decomposition** occurs when one or multiple parts are depicted to represent the whole, such as a microscope for the concept of *laboratory*.

(e) **Temporal decomposition** is a useful strategy for representing events and verbs. This category includes the snapshot-like pictographs of the various sports created for the Olympic Games.

(f) **Body language** concerns the depiction of facial expressions, body postures, and gestures. A smiling face can represent *happiness* and *enthusiasm*, among other things.

(g) **Metaphors** connect concepts by similarity, such as a pictograph featuring a majestic eagle to represent the king, or a puzzle to represent the mind of a person with an autism spectrum disorder.[6]

(h) Representation by **contiguity** is analogous to metonymy in language. For instance, a sphygmomanometer is a medical instrument that is commonly used to represent *blood pressure*, while an insulin pump represents *diabetes*.

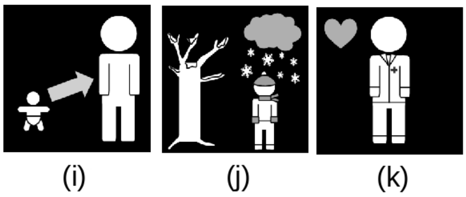


Figure 1.9: Additional indexical relations: (i) *to grow (up)*, (j) *winter*, and (k) *cardiologist*.

However extensive this classification may be, it is not yet exhaustive. Based on our observations of the sets that we use in the translation technologies, we propose to add a number of subtypes. Sclera examples are shown in Figure 1.9.

(i) **Change of state** is a strategy for depicting physical or psychological changes which occur when an object or a person crosses from one state into another. An example of this is the process of *growing (up)*, which is represented by means of its initial state (*baby*) and its final state (*adult*).

(j) **Associative exemplification** is similar to exemplification, but uses semantically related examples for depicting abstract concepts. For instance, *winter* can be depicted by means of snow, a person wearing warm clothes, and a tree without leaves.

(k) **Subcategorisation** occurs when a secondary pictograph is added to further specify certain properties of the main pictograph. For example, a regular *doctor* can be represented as a person wearing a lab coat. In order to differentiate between various types of doctors, then, a second element can be added for clarification purposes. As such, the *cardiologist* pictograph depicts a doctor and a heart, while a *pediatrician* is represented by means of a doctor and a baby.

[6]Of course, these pictographs could also be read iconically. An eagle pictograph on the map of a zoo would most likely refer to the bird, instead.

1.1.3.3 Symbolic Relations

The relation between a pictograph and a referent can also be arbitrary or *symbolic*, in which case the connection is fixed by convention. Nakamura & Zeng-Treitler (2012:543-544) identify three subtypes of representation by arbitrary convention. Examples are shown in Figure 1.10.

(a) **Abstract** symbols are geometric or verbal. On food packaging, the pictograph of the three arrows stands for *recycling*. In traffic, the blue pictograph with the letter *P* stands for *parking*. Traffic signs make use of fixed shapes, borders, and colours that are assigned a fixed meaning, which must be carefully studied before one learns how to drive.

(b) **Concrete** symbols resemble a physical object. The pictograph of a skull on a bottle of detergent indicates that the liquid is poisonous. The caduceus is a mythological staff that now represents medicine. It is up to the reader, who may or may not be familiar with the symbolic meaning of the images, to decide whether these pictographs should be read iconically or symbolically.

(c) In the case of **transposed** symbols, the convention lies in the referent, rather than in the pictorial representation. One such example is the flexing of one's arm, which is conventionally associated with the concept of *strength*.

(a) (b) (c)

Figure 1.10: Sclera examples of symbolic relations: (a) *recycling*, (b) *blood*, and (c) *strong*.

Symbolic pictographs are not very common in pictograph sets that are designed for people with ID. However, iconic and indexical pictographs are often combined with symbolic elements. Pictographs sometimes contain common symbols such as crosses, which express a negation or prohibition, or arrows, which shift the user's attention toward a specific area of the pictograph.

1.1.4 Conclusion: Semantic Properties

From a Peircean point of view, pictographs can be icons, symbols, or indices, depending on the way in which the relation between their visual representation and the intended referent is interpreted, with many examples of all three subtypes to be found everywhere in our daily lives and within pictograph sets that are designed for people with ID. However, while all pictographs are signs, the inverse relation does not hold. Other (even more) common examples of signs are words. Just like pictographs, words can be iconic (onomatopoeia), indexical (when pointing directly to their meaning within a given context, such as *you* or *now*), or symbolic (most words). Moreover, we can think of a number of other iconic signs, such as photographs or realistic paintings, which we would not consider to be instances of "pictographs" at all. Where lies the difference, then? There is clearly much more to be said about the specific communicative functions and formal characteristics of pictographs, allowing us to distinguish them from other signs. We will explore these questions in the following sections.

1.2 Function: Communication Through Pictographs

Pictographs are typically used to comply with a wide range of communication needs. Section 1.2.1 focusses on the communicative role of pictographs in three domains: daily life, chat or text messages, and Augmentative and Alternative Communication (AAC). It is often argued that pictographs gain their communicative power from their universality. In section 1.2.2, we debate whether pictographs can, indeed, always be understood correctly across language barriers. Section 1.2.3 concludes.

1.2.1 The Advantages of Using Pictographs for Communication

In a number of situations, the use of pictographs may be preferred over written text. Section 1.2.1.1 describes the advantages of pictographs in traffic, workplaces, and locations where people from different cultures gather. The use of emoticons for nonverbal communication is discussed in section 1.2.1.2. Finally, section 1.2.1.3 shows that people with writing and/or reading difficulties may benefit from using images that are specifically designed for AAC. The translation technologies that are described in the following chapters make use of the latter type of pictographs.

1.2.1.1 Pictographs in Our Daily Lives

Where people from around the world gather, pictographs can be found. This section discusses two essential communicative functions of pictographs in our daily lives, namely their potential to cut across language barriers, and their ability to communicate large amounts of information quickly and concisely (Isherwood et al. 2007:465).

1.2.1.1.1 Crossing Language Boundaries Pictographs are a consequence of globalisation (Vaillant & Castaing 2008:2). With tourists and businessmen travelling all around the world and goods being bought and sold everywhere, there is a dire need to address a great number of people with different linguistic backgrounds, ideally, in a uniform manner. Two early examples are the pictographs on the map of the suburban timetables of the London and North Eastern Railway, designed by George Dow in 1936 to indicate facilities available at or near each station, and the pictographs that were used in the Tokyo Olympic Games in 1964 (Zhou 2014:162). Since then, pictographs have appeared at every venue where people with different cultural backgrounds gather, such as airports or train stations (see Figure 1.11[7]).

Figure 1.11: American Department of Transportation pictographs: *escalator*, *nursery*, and *ground transportation*.

The idea of using pictographs to convey messages to those who do not speak the language at hand is even believed to be so strong, that plaques with pictographs are attached to the bodies of the Pioneer and Voyager space ships. In the (unlikely) event the space ships are intercepted by extraterrestrials during their missions to explore beyond the solar system, the pictographs are supposed to provide help in explaining the origins of the explorers (Cowgill & Bolek 2003:5).

There are many reasons why pictographs may be preferred over written text. Leemans (2001:12) stresses that visual languages have the ability to overcome some of the problems that written languages encounter. Some of the disadvantages of learning

[7] Source of image: https://commons.wikimedia.org/wiki/File:DOT_pictograms.svg

natural languages to write in are their varieties, their inconsistencies and irregularities, and the fact that there exist often many different ways to say the same thing. The inconvenience of translation can be avoided by means of an international auxiliary language with a restricted grammar and a limited vocabulary. The choice for a pictograph language is then motivated by several reasons. Fitrianie & Rothkrantz (2005:574) highlight the parallel temporal and spatial configuration of visual languages, similar to sign languages, in opposition to the linear ordering of words. Furthermore, pictographs use little space for representing a lot of information (Böcker 1996:108). Tanimoto (1997:5) remarks that visual languages are easy to learn in comparison with textual languages, especially when the language objects are able to "explain themselves". This is strong evidence for the clarity, unambiguity, and usefulness of visual languages.

However, while pictographs are already widely spread around the world, their potential to cut across language barriers has not yet been exploited to the fullest. We present two scenarios: humanitarian aid and medical consultations.

Non-profit humanitarian associations bump into communication barriers when delivering information in the language of the affected populations, and rely heavily on the efforts of skilled translators.[8] Humanitarian organisations have to provide vital information to people in need, in a language they can understand. All too often, there is an urgent need for translator volunteers. For the reasons mentioned above, pictograph-based systems could provide a safety net in situations where urgent communication matters arise.

Somers (2007) found that communication between clinicians and patients with limited proficiency on the majority language could benefit from using pictograph-supported communication devices, as an alternative to using automatic speech recognition or regular machine translation techniques. This is especially the case for so-called "under-resourced" languages. Although high satisfaction and understandability ratings were achieved, Somers remarks that his image-supported device did not yet allow the users to go off-script, with only a handful of fixed questions and answers available for the doctor and patient to choose from. Yet, this study greatly favours the idea of encouraging pictograph-based communication within medical settings. This finding is echoed by Wołk et al. (2017), who developed a smartwatch application that offers its wearer a limited amount of pictographs that are designed specifically for emergency medical applications.

[8] See, for example, Translators without Borders: https://translatorswithoutborders.org/

While these situations go beyond the scope of this dissertation, future work in this domain is desirable.

1.2.1.1.2 Calling Attention Beside their potential to cut across language barriers, pictographs can serve as "instant reminders" of a hazard or an established message (Tijus et al. 2007:18). Pictographs are more robust to changes in scale, reading speed, and distance than text (Fitrianie & Rothkrantz 2006:535). As pictographs are visually more distinctive than words, they are more easily recognised (Böcker 1996:108) and have the ability to evoke readiness to respond for a fast exchange of information (Fitrianie & Rothkrantz 2005:574). Images can be processed in parallel and are more easily perceived in suboptimal conditions than words, which require serial processing (Tijus et al. 2007:23). This explains why pictographs are used in traffic, instead of words. In 1909, the first four road signs showing hazardous road conditions (bump, road crossing, curve, and railroad crossing) were introduced at the Convention on the International Circulation of Motor Vehicles (Cowgill & Bolek 2003:4). This gradually expanding system was adopted by several European countries. The United States continued to use a sign set consisting mainly of English words until 1970.

As pictographs have the ability to focus one's attention and facilitate memorisation, they also present a number of advantages for clinicians, who are expected to take rapid medical decisions. Textual documents consisting of words and numbers may not always be the most efficient way to convey the information required for drawing valid conclusions concerning a patient's health issues, since time is limited during consultations (Lamy et al. 2008:2). In order to facilitate the access to drug monographs, a system of medical pictographs could be used. The VCM (*Visualisation des Connaissances Médicales*) language, for instance, consists of a small number of graphical primitives and combinatory rules and represents the various signs, diseases, physiological states, life habits, drugs, and tests described in monographs. VCM aims to complement medical texts (rather than replacing them) by highlighting pieces of text or helping physicians find the desired parts of the text (Pereira et al. 2014:3). It should be noted, however, that many diseases are difficult to represent graphically (see also section 1.1.3.1 on the topic of iconicity).

1.2.1.2 Emoticons: Enriching the Emotional Expression

Pictographs can also be used to augment text messages with emotional marking, in which case they are referred to as *emoticons*. Emoticons were originally combinations of ASCII characters such as *:-)* (Western use) or ^_^ (Japanese use) for a happy face (also called a *smiley*). They are now often replaced by graphic representations.

Emoticons are used on personal computers and phones to enrich the emotional expression or nuance of words (Takasaki 2006:280). They compensate for the lack of paraverbal communication cues, such as facial expressions, intonation, gestures, and other bodily indicators (Skovholt et al. 2014:781). Emoticons can express a wide range of emotions, such as laughter (by means of a smiling face with tears),[9] sadness (by means of a crying face), or love (by means of a heart-eyed face), among other things (see Figure 1.12[10]).

Figure 1.12: Emoticons: *laughter*, *sadness*, and *love*.

1.2.1.3 Pictographs for AAC: Enabling Communication for and with People with ID

The use of images as a means of conveying information to people with reading and writing difficulties is not an invention of our time. In the Western European Middle Ages, when the language of the Church had become unintelligible to the masses, frescoes and stained-glass were used to tell stories from the Bible (Vaillant & Castaing 2008:1-2). Emblem books, such as Albrecht Dürer's *kleine Passion* from 1510, were composed only of woodcut prints. And shops or workshops were often indicated by hand-carved or metal-forged signs, like a boot for a shoemaker or a key for a locksmith (Dreyfuss 1972:38).

Literacy is described as "the ability to read and write" (Maiti et al. 2011:172). The literacy all over the world is estimated at 84%. Approximately 54.74% of the total population in the developing countries is considered literate. Cowgill & Bolek (2003:4-5) note that nations with high illiteracy rates, such as India or Egypt, still

[9]Remarkably, Oxford Dictionaries declared this emoticon to be the 2015 word of the year. Source: http://time.com/4114886/oxford-word-of-the-year-2015-emoji/

[10]Source of images: https://commons.wikimedia.org/wiki/File:Emojione_(1F602.svg, 1F622.svg, and 1F60D.svg)

rely on the use of images in everyday matters. For instance, political parties and candidates must assign themselves an image (such as a car, a book, or a clock, which often symbolise a deeper meaning) in order to make it easier for illiterate voters to identify their choices on the ballot.

While having no access to education is, of course, an important cause of illiteracy, reading and writing difficulties may also arise as a result of an intellectual disability. An intellectual disability affects one's capacity to think, from conceptualising to remembering and understanding written text (Keskinen et al. 2012:374). For people with ID, written communication is often only possible through the use of graphic representations. It is estimated that between 2 and 5 million people in the European Union could benefit from the use of pictographs as a substitute or form of support for written text (Keskinen et al. 2012:375).

These pictographs, which are often specifically designed for people with ID, are a form of Augmentative and Alternative Communication (AAC). AAC includes both unaided systems, such as body language and sign language, and aided systems, such as communication books and electronic communication aids, allowing people to build messages that enable them to participate more fully in their social roles and activities, including interpersonal interaction, learning, and education (Tuset et al. 2010:797-798).

Pictographs enable a wide vocabulary, aid in concept development, permit interaction within daily environments, and assist in socialisation. Images reduce the demand on the user's cognitive skills, can be individualised based on the person's needs, and support acquisition of communication through vision. If words and phrases are used, there is a cognitive, physical, and visual load involved in remembering and accessing the individual words and phrases (McCoy 1998:2). It is thus not surprising that many schools and institutions for people with ID rely on pictographs for daily communication.[11]

In order to use pictographs effectively, individuals with complex communication needs must learn their meanings, both alone and in combination, and how to produce

[11] Another type of pictographs for AAC are tangible pictographs. More than 60% of people with visual impairments struggle to develop symbolic communication and may even never acquire the skills needed to express their most basic needs and wants (Ivy et al. 2014:474). For these users, tangible pictographs can be used. Tangible pictographs are three-dimensional cards embedded with whole or partial objects to represent a person, place, activity, object, idea, or action. Visual supports must be replaced with auditory, tactile, or olfactory equivalents. For instance, a hard plastic cup represents a drink, an empty bag of crisps represents a snack, and a small metal bell represents music.

them in communicative contexts (Martínez-Santiago et al. 2015). There are several approaches to teaching pictograph-based communication, which are commonly referred to as "language modelling techniques". One such popular technique is the System for Augmenting Language (SAL) (Romski et al. 2009). It is intended to introduce the use of pictographs by observing the way in which they are used by the beginning communicator. In SAL, the instructor teaches the beginning communicator to associate the meaning of a given word with the pictograph and the sound of such a word (Martínez-Santiago et al. 2015:159), for instance, by pointing at the *play* and *outside* pictographs while uttering the sentence *let's go outside and play*.[12] In more sophisticated models, the individual with complex communication needs learns how to make his/her own requests and how to establish topics of conversation, for instance, by means of conversation books or AAC displays.

Not only people with ID benefit from using pictographs for communication. Pictographs are also used by people with limited motor skills, allowing them to communicate more effortlessly. On touch screen devices, pictograph selection requires less effort than letter-by-letter word construction; a user could request a glass of water with a single tap, whereas spelling out the word *water* would require five taps. Input methods vary from finger pointing to tongue input devices and eye-gaze input systems. The description of these devices is beyond the scope of this dissertation. Among the pictograph sets that permit people with a motor disability to communicate, who would otherwise be cut off to express themselves (Tanimoto 1997:4), are *Blissymbols* (Bliss 1965) (see section 1.3.2.1.1) and *Minspeak* (Baker 1982a).

The following chapters describe a pictograph-based technology for AAC that uses Sclera and Beta pictographs (see section 1.3.2.3), two sets that are commonly used in schools and day centres for people with ID in Flanders.

1.2.2 The Universality Issue

In section 1.2.1.1, we discussed the potential of pictographs to cut across language barriers. However, whether it is truly possible to attain the ideal of creating a completely universal pictograph language is a topic that has often been debated in the literature. Since each interpretant is unique and possesses a certain cultural and social bias, the way in which the object is recalled by the sign can be very different for

[12]Simple signs and gestures can even help pre-verbal children without complex communication needs to understand concepts and to communicate before they can talk.

each person (Goonetilleke et al. 2001:742).

Successful communication can only be realised when two participants share a common interpretation (Cho et al. 2007:222). However, the understanding of pictographs may differ across different cultural groups (Munemori et al. 2010:474). Differing interpretations can lead to misunderstandings (Soares 2015:66).

It is, therefore, a false assumption that a pictograph will necessarily be understood by an illiterate or even by a literate person (Cowgill & Bolek 2003:22). For example (Cowgill & Bolek 2003; Munemori et al. 2010; Leemans 2001), the cow is a source of nourishment to westerners who drink milk and eat its meat, while Indians consider the animal as an object of veneration. The Eiffel Tower cannot always easily be identified by Japanese people, who might mistakenly recognise it as the Tokyo Tower. Sharp objects symbolise *cutting short good luck* to Chinese, Japanese, and Vietnamese people. White signifies *death* in some cultures. In many Asian countries, people do not touch each other in public. Making a circle with your index finger and thumb means *OK* in the USA, but symbolises *money* in Japan. Western countries use forks, Eastern countries use chopsticks. And not even pictographs of public toilets share a universal design.

In other words, all beliefs, customs and ethics are relative to an individual within that individual's cultural and social context (Cho et al. 2009:173). The principle that a person's beliefs, values, and practices should be understood based on that person's own culture is known as *cultural relativism*. Any difference of cultural context can entail differences in interpretation (Vaillant & Castaing 2008:7).

New cultural productions tend to be interpreted within the filtering patterns that are passed on by pre-existing cultural objects. For instance, the pictographs for *man* and *woman* that are often used in international public information signs show a person wearing trousers and a person wearing a dress. These pictographs are already lagging a couple of decades behind our present cultural habits; Western European women are often seen wearing trousers. On top of that, they are completely irrelevant in cultures where men wear ample gowns.

It is thus advisable that developers of publicly available pictograph sets consider the impact of culture as they develop future products, as the perceptions of meanings are likely to vary as a function of culture or ethnicity (Huer 2000:184). The idea of creating a universal set that can be understood unambiguously across all possible cultures is most likely an impossible goal to achieve. Rather, pictograph developers should consider targeting a specific culture or a subset of cultures.[13] This idea is nicely demonstrated by Draffan et al. (2015), who aim to generate acceptable core vocabularies and pictographs for Arabic AAC users. Based on a collection of word lists used by the users, their families, teachers, and speech and language therapists in Qatar, the authors construct a new pictograph set for Arabic users. The set was evaluated by a group of target users, who were presented with a total of four voting criteria (feelings about the image, representational power of the image, colour contrast, and cultural sensitivity). This way, an appropriate pictograph set for Arabic AAC users - but not necessarily for users who originate from other cultures - could be constructed. In the resulting set, for example, men are wearing the traditional attire of the Arabs.

Another solution toward bridging the cultural barrier is shown by Yuizono et al. (2012), who present a method for improving semantic expressions in pictograph chat conversations between Japanese and Chinese speakers through the use of a "semantic pamphlet", a vocabulary list, which assists the users' comprehension of the pictographs. Of course, this method is not appropriate for people with reading difficulties.

1.2.3 Conclusion: Communication Through Pictographs

We discussed the communicative functions of pictographs. Pictographs replace or support written instructions in places where people with different linguistic backgrounds gather or in situations where information must be processed quickly, such as in traffic or during medical consultations. Emoticons enrich the emotional expression in situations where paraverbal and gestural information is absent. Finally, we discussed how people with reading and/or writing difficulties can benefit from using a coherent set of images as a means of written communication.

We have argued that pictographs are not necessarily as universal as they are often claimed to be. Cultural differences can easily lead to wrong interpretations. Rather than searching for abstract metarepresentations that show no cultural bias whatsoever, graphic designers and pictograph developers should focus on creating pictograph sets that are appropriate for well-defined communities.

[13]For instance, one might want to tailor a pictograph set toward the specific needs of people with ID in Flanders, and use the emblems of their regional services, such as the logo of the national post service, which are more recognisable to the end users, rather than using "generic" pictographs.

1.3 Form: The Formal Characteristics of Pictographs

Pictographs are designed according to a number of design principles, although these principles can vary across sets. In section 1.3.1, we discuss a number of general recommendations for pictograph design, while highlighting some of its most important formal characteristics, and uncovering the discrepancies between different theories. Section 1.3.2 focusses on the particular design principles of five specific pictograph sets: two for cross-cultural communication, one for enriching the emotional expression, and two for AAC purposes. We conclude in section 1.3.3.

1.3.1 Design Principles that Apply Across Sets

A thorough analysis of the formal characteristics of pictographs leads us into the domain of graphic design. While pictographs have been around since the beginning of the 20th century, it was Otl Aicher, the lead designer of the 1972 Summer Olympics in Munich, who became one of the first graphic artists to conceive a set of design principles. In order to aid foreign visitors and athletes in finding their way around the stadium, Aicher created a set of pictographs featuring the different Olympic sports according to the following six principles (Cowgill & Bolek 2003:12):

- Pictographs should not look illustrative.
- They should not be particular to a specific culture.
- They should avoid cultural taboos.
- They should be easily readable and understandable.
- Their design should be based upon uniform rules.
- The educational level of the viewer should not be a factor in their understanding.

Later accounts of pictograph design research have been drawing inspiration from a number of well-established psychological theories, such as Gestalt psychology, a movement that took off in Berlin in the 1920s. According to Gestalt Theory, the stronger the relationship between elements on a page, the better the communication.

Many of the design principles discussed below derive from Gestalt Theory, which aid in creating unity in the whole of a design.[14]

Firstly, since pictographs belong to sets, they must be **similar enough** in order for us to perceive them as belonging to the same set.[15] Cowgill & Bolek (2003:6) and Tijus et al. (2007:31) underline the importance of uniformity in design throughout the individual pictographs and throughout the entire system. Overall consistency allows people to learn new pictographs that belong to the same collection more effortlessly (Beardon et al. 1992:28). In the case of traffic signs, for instance, clear similarities can be identified throughout the whole set. For example, most prohibition signs have a red border and the shape of a circle. However, it is also important to remark that the pictograph should be different enough from other pictographs that belong to the same set, in order to avoid confusion.

Secondly, pictographs are a **stylised** representation of reality. However, the amount of detail in pictographs needed in order to convey meaning is disputable. While there must be appropriate levels of complexity and detail to maximise comprehension (Tijus et al. 2007:29), pictographs should also be as visually simple as possible, containing only the essential facts about the concept that is depicted (Cowgill & Bolek 2003:6). Although it is often believed that detailed depictions make it easier for users to access their existing knowledge about the items or events that are represented (Isherwood et al. 2007:466), other theories (Byrne 1993; Scott 1993) claim that simpler icons

[14]Among these design principles are proximity, similarity, continuity, closure, and figure/ground, which we briefly discuss here (Bigman 2014; Busche 2015; HOW Staff 2015):

- *Proximity*: If there is a series of objects, we tend to perceive objects that are close together as a group.
- *Similarity*: We perceive elements as belonging to the same group if they look like each other. The principle of similarity can be triggered using color and size, among other things.
- *Continuity*: According to the principle of continuity, once the eye begins to follow something, it will continue traveling in that direction until it encounters another object.
- *Closure*: When we see a figure that appears to be partially hidden, our tendency is to complete it, even if that means that we need to supply imaginary visual information.
- *Figure-ground*: This principle is based upon the relationship between an object and the surrounding space. The figure is the element in focus, while the ground is the background behind the figure.

[15]We will reserve the term *pictograph language* for the combination of individual pictographs, that belong to a specific *pictograph set*, into a larger meaningful structure.

can be discriminated more easily and are easier to locate in visual search. Isherwood (2009:200) remarks that users respond more quickly and accurately to concrete pictographs than to abstract pictographs, thus supporting the idea that a pictorial or visually obvious image will be more easily understood by the user. However, such performance advantages diminish over time when users are allowed to gain experience with a set of pictographs. Familiarity has longer term effects in determining response times, because of familiar items being easier to access in the long-term memory.

Finally, **visual information content must be maximised** in order to facilitate recognition. Silhouettes or side views often contain more useful information than frontal views (Cowgill & Bolek 2003:6). A chair, for instance, should not be depicted from above (Leemans 2001:143). 3D representations are preferable when the object has three dimensions (Tijus et al. 2007:31). Furthermore, in digital environments, using multiple pictures or animated graphics is the best way to convey motion or contact verbs, rather than showing single, static images (Ma & Cook 2009:364).

Bad design principles may lead to issues on the syntactic, semantic, or pragmatic level (Leemans 2001:136-139). If the elements of the visual representation do not combine properly (see Figure 1.13), the sign is not coherent and, therefore, it is confusing. When secondary elements are too strong, the most important information in the image could be difficult to extract. Furthermore, Leemans strongly advises against using obscure or strained metaphors or pictographs that are based on inside jokes, figures of speech, slang, or insults.

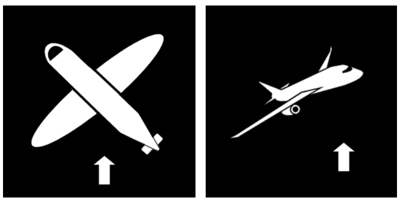


Figure 1.13: On the left-hand side: Example of a badly designed pictograph, depicting a plane's *take off*. The plane is depicted from above, while the arrow seems to be pointing at or pushing the plane's tail - at least, if the viewer assumes that all elements exist within the same coordinate system. On the right-hand side: Example of a pictograph in which all elements exist within the same coordinate system.

1.3.2 Set-Specific Design Principles

While the previously discussed general design principles apply to all pictographs, individual sets are often characterised by their own set-specific design guidelines. In analogy with the three communicative functions of pictographs discussed in section 1.2, we present two noteworthy sets that are developed for intercultural communication (section 1.3.2.1), one emoticon set (section 1.3.2.2), and two sets that are designed for people with ID (section 1.3.2.3). The latter sets, Sclera and Beta, are used in the Text-to-Pictograph and Pictograph-to-Text translation systems that are described in the following chapters.

1.3.2.1 Pictograph Sets that Aim at Facilitating Intercultural Communication

This section presents two sets that are primarily conceived to facilitate intercultural communication: Semantography (a highly symbolic set) and VIL (a highly iconic set).

Figure 1.14: Semantography (Blissymbols) translation of *I like to go to the cinema*.

1.3.2.1.1 Semantography *Semantography* was initially developed by Bliss (1965) with the objective of crossing language barriers among people, especially scientists, all over the world (Tanimoto 1997:4). The system was later adopted for AAC purposes under the name of *Blissymbols*.

The Semantography pictographs are composed from graphical primitives (see Figure 1.14[16]). These primitives are basic objects in the world, such as *house* or *eye*, which can be combined to form more complex pictographs, such as *cinema*. There are two types of primitives: those that present a physical thing using an outline, such as a horizontal line for *earth*, and those representing non-physical things using geometric symbols, such as arrows for directionality (Leemans 2001:54). The grammar

[16] Source of image: https://commons.wikimedia.org/wiki/File:Bliss_cinema.png

of Semantography is linear and similar to the word order of most Western European languages (i.e., SVO).

Semantography's decomposable pictographs combine in a consistent and logical way. Users can use the primitives to generate new pictographs. However, the system is highly symbolic. It not very intuitive and, therefore, it may take a long time to learn.[17]

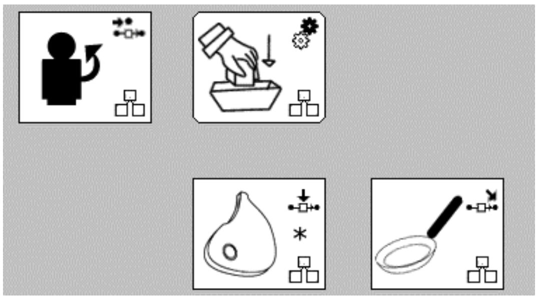


Figure 1.15: VIL translation of *I put (past tense) two pork chops in a frying pan*. The past tense and the numeral are modifiers of the verb and noun pictographs, respectively. They are not shown on this image, as the user must click on the pictographs in order to see them. The syntax of VIL is two-dimensional: The predicate (verb) has its subject on the left-hand side, and its direct object appears directly below it. Example taken from Leemans (2001:211).

1.3.2.1.2 VIL: A Visual Inter Lingua Pictograph design in *VIL: A Visual Inter Lingua* (Leemans 2001:139-147) is based on a very elaborate set of guidelines, which "allow the images to be recognised at a glance" (see Figure 1.15).

Leemans adheres to the view that simple pictographs, which represent less visual information and therefore require less conscious effort, can be recognised more easily than complex ones. Simplicity is reached by reduction or removal of elements that are not essential for the communication task at hand.

Leemans notes that pictographs are part of a larger system. Therefore, the individual elements must work together effectively to attain cohesiveness. This is done, for instance, by repetition of common forms throughout the set. Visual cues, such as scale, contrast, and proportion are applied in a consistent way.

Furthermore, Leemans emphasises that the effectiveness of the depiction of a particular object or event is largely determined by viewpoints, cultural background, and

[17] In Chapter 11, we will present a number of communication technologies that make use of this set.

fashion. The viewpoint exploits the viewer's familiarity with a particular, characteristic perspective of an object. The viewer's cultural background or understanding of the world influences the way in which a pictograph is perceived (see section 1.2.2). Lastly, fashion is time-bound. The floppy disk pictograph, which has been used as a "save" button on computers for many years, is outdated, as diskettes have become obsolete nowadays. Based on a survey held in 2013,[18] only 14 percent of children knew what the save button actually represented. Ideally, pictographs are future-proof and have a long life span.[19]

1.3.2.2 Pictograph Sets that Enrich the Emotional Expression

One widely used set of graphic emoticons is called *emoji*, a series of standardised pictographs that are built into handsets. Many emoji originate from Japanese culture. Today, ninety-two percent of the online population uses emoji, sending 6 billion images a day through text, email, and social media platforms.[20]

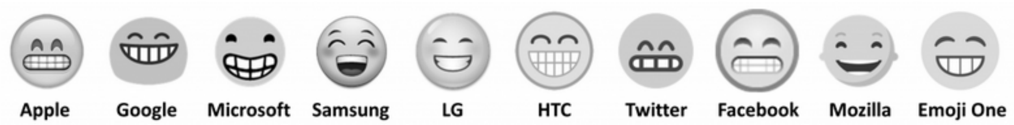


Figure 1.16: Different renderings of the *happy* emoji. The graphics look different for each platform. Example taken from Miller (2016).

Emoji render differently on different viewing platforms. Emojipedia,[21] a website serving as an encyclopedia for emoji, lists 17 such platforms, meaning that there may be at least 17 different renderings for a given Unicode emoji character (see Figure 1.16). Miller et al. (2016) explored whether emoji are consistently interpreted, as well as whether interpretation remains consistent across renderings by different platforms. For five different platform renderings of 22 emoji Unicode characters (standalone versus in context), the authors found disagreement in terms of both sentiment and semantics, and these disagreements only increased when considering renderings across platforms.

[18]http://www.teachhub.com/first-infographic-made-kids-national-library-week

[19]For more information on the VIL pictograph interface and the VIL syntax, we refer to Chapter 8 and Chapter 11.

[20]https://sciencenode.org/feature/the-not-so-secret-language-of-.php

[21]https://emojipedia.org/

1.3.2.3 Pictograph Sets for People with ID

The previously discussed sets were not designed for users with ID. Pictographs for cross-cultural communication are often complex and highly abstract, and their use sometimes requires a solid knowledge of grammatical properties. Emoji, on the other hand, are too small, and for many users with a disability, there are simply too many images with very slightly nuanced details to choose from. Furthermore, emoji vary across devices and platforms, making it difficult for users with ID to familiarise themselves with the pictograph dictionary at hand.

Over the years, a number of pictograph sets have been developed with the specific objective of enabling easy communication for and with people with ID within an offline setting, such as schools, day centres, or sheltered workshops. In these pictograph sets, many characteristics of natural languages are absent. For instance, in the majority of sets, no distinction between singular and plural is made. Tense, aspect, and inflection information is removed, and there are no auxiliaries and articles (with some exceptions). The pictograph translation systems that are described in this dissertation currently give access to two pictograph sets that were designed for AAC: Sclera[22] and Beta.[23]

Our choice for these two sets is mainly motivated by the fact that they are developed in Flanders,[24] and the fact that they are both used in the WAI-NOT environment (see section 2.1). Moreover, the Sclera pictographs are open-source, whereas a licence for the coloured Beta pictographs can be obtained for a reasonable fee. The black-and-white Beta pictographs are available for free.

Different pictograph sets are used across different communities or cultures. For instance, in Spain, the ARASAAC set[25] is widely spread among users with reading and writing disabilities. Picture Communication Symbols[26] are popular in the USA, but they are not adapted to other cultures, whereas the Flemish non-profit organisation De Rand[27] proposes its own pictograph set to meet the complex communication needs of immigrants in Belgium.

---

[22]http://www.sclera.be/

[23]https://www.betasymbols.com/

[24]Sclera, for instance, uses some logos of national institutions, making the pictographs more recognisable for Flemish users.

[25]http://www.arasaac.org/

[26]http://www.mayer-johnson.com/education/symbols/pcs/

[27]https://www.derand.be/

1.3.2.3.1 Sclera The *Sclera* set[28] comprises over 13,000 pictographs that have been specifically designed to meet the communicative needs of people with ID. More pictographs are added upon user request. Visually, they are characterised by their uniformity, with most pictographs adhering to a strict black-and-white color scheme. There are, however, some exceptions, such as the denotation of prohibition (red background or red cross) or permission (green background), and the depiction of colours, such as *purple*.

Sclera's high contrast and lack of distracting details has proven to be very popular among people with an autism spectrum disorder, as opposed to photographs (Noens & Berckelaer-Onnes 2002:220). Certain groups of users vary in their preference for a particular depiction of a concept, finding one pictograph more obvious than others. In the Sclera set, therefore, there are often multiple alternative pictographs available for a given concept (see Figure 1.17).

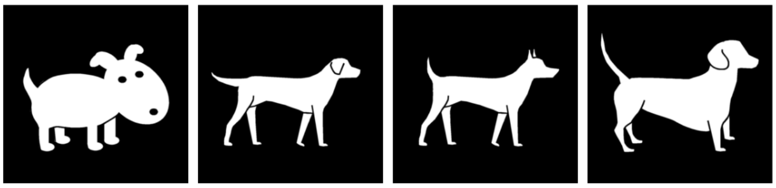


Figure 1.17: There are four different pictographs available for the concept *dog*. The Text-to-Pictograph translation tool only uses one variant.

Most Sclera pictographs correspond to nouns and verbs. The pictographs generally do not distinguish between singular and plural readings, and they do not encode tense or inflection. Articles do not exist in the Sclera set.

In addition to the depiction of many *simplex* concepts, Sclera pictographs often represent *complex* concepts (see Figure 1.18), like a verb and its object (such as *to feed the dog*) or phrases (such as *salt and pepper*). There are hardly any pictographs for adverbs or prepositions.

On a thematic level, the Sclera set contains a wide variety of pictographs, relating to independent living (for example, bank cards, ATMs, or taking the train), behaviour, culture, contemporary lifestyle and technology, as well as a number of potential taboo subjects, such as sexuality, religion, hygiene, and death.[29] The characters that are

[28]Freely available under Creative Commons License 2.0.

[29]The taboo pictographs aid caregivers and family members in illustrating unacceptable behaviour. Uncensored pictographs are particularly helpful for people with a mild intellectual disability. (Source:

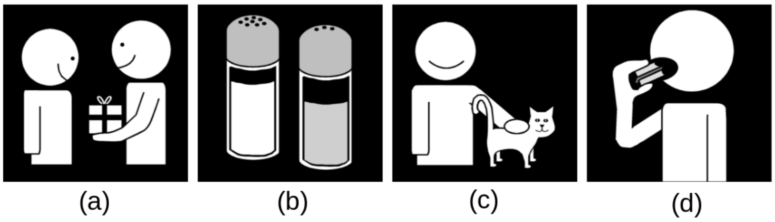


Figure 1.18: Examples of complex Sclera pictographs: (a) *to give a present*, (b) *salt and pepper*, (c) *to stroke a cat*, and (d) *to eat a sandwich*.

depicted on the pictographs have no specific race,[30] body type, age, or gender, thus referring to virtually any person in the world (with the exception of characters that appear on pictographs that relate to race, body type, age, or gender; think of concepts such as *multiculturalism* and *racism*, *overweight* and *thin*, *grandfather* and *child*, or *mother* and *girl*) (see Figure 1.19).

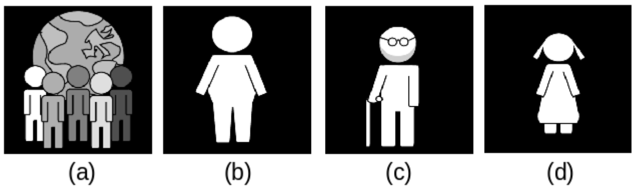


Figure 1.19: Examples of Sclera pictographs that explicitly refer to race, body type, age, or gender: (a) *multiculturalism*, (b) *overweight*, (c) *grandfather*, and (d) *girl*.

1.3.2.3.2 Beta The *Beta* set consists of more than 3,000 pictographs. Easy recognition being one of the main objectives, Beta is characterised by its overall consistency and the use of different types of arrows and dashes (pointing to an object, indicating changes in space or time, or depicting actions, in a similar way to comic strips and cartoons).

The pictographs in this set make much more use of colour than those in Sclera, which makes them visually more appealing for some people. However, Beta pictographs depict nearly always *simplex* concepts. Therefore, the complex Sclera pictographs that were shown in Figure 1.18 would correspond to at least two Beta pictographs each.

Brabants Dagblad, March 26, 2013 - http://www.sclera.be/resources/img/vzw/pers/brabantsd.pdf)

[30] Although the characters are white, nonetheless.

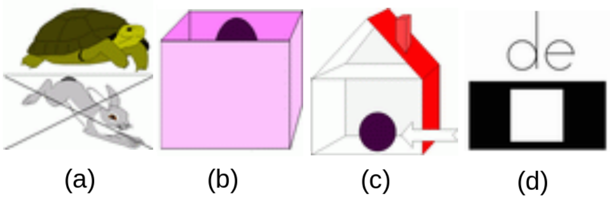


Figure 1.20: Examples of Beta pictographs: (a) *slow* (adjective), (b) *in* (preposition), (c) *inside* (adverb), and (d) *the* (article).

While the Beta set is smaller than the Sclera set, it contains a larger variety of adjectives, adverbs, prepositions, and even articles (see Figure 1.20). On the other hand, there are less pictographs related to behaviour or taboo subjects, with most of the Beta images referring to everyday objects or feelings. The main character depicted on most Beta pictographs is a white male wearing a purple shirt. As such, they are less universal than Sclera pictographs.

1.3.3 Conclusion: Formal Characteristics

This section discussed the formal characteristics of pictographs. Although many design guidelines have been proposed over the years, we were able to infer a number of common traits. First of all, while the level of detail that is required in pictographs is not always agreed upon, it can be concluded that the images are nearly always stylised versions of reality, as opposed to photographs or realistic drawings. The referents are depicted with fewer lines and flat colours (i.e., no shading) or no colours at all, and accessory information, such as backgrounds, is removed. Secondly, pictographs belong to sets, and uniformity in design throughout the entire system is essential. Uniformity aids a pictograph user in deriving the meaning of an unknown pictograph that belongs to a known set with more ease. Finally, visual information content on pictographs is maximised in order to facilitate recognition. However, as was shown in the previous section, prototypicality can be culture-bound.

1.4 Conclusion: Toward a Definition of Pictographs

In this chapter, we shed light on pictographs from three different perspectives: meaning, function, and form. Section 1.1 discussed the semantic relations between pictographs and their intended referents. In section 1.2, we showed that pictographs play an important role in everyday communication, for people with and without a disability. Finally, section 1.3 highlighted some of the most important formal characteristics.

Speaking in Peircean terms, pictographs can be iconic (physically resembling the entity they stand for), indexical (having an indirect relation with the entity they stand for), or symbolic (having a purely conventional connection with the entity they stand for) representations of concrete objects or abstract concepts. They differ from other visual representations, such as realistic drawings and photographs, in that they are stylised representations of reality - a principle that results in the removal of non-essential visual elements - and belong to a larger collection of images that have been designed according to specific design principles. The use of pictographs as a substitute or support for written text allows for quick processing of information and communication with large groups of people with different linguistic backgrounds, although true universality still remains an issue. Within the context of AAC, pictographs allow people with communication difficulties to express themselves and access information more easily, as there is less physical and cognitive load involved in accessing and remembering the images.

The pictograph translation systems that are presented in the following chapters are primarily conceived as tools that allow people with ID to participate in the digital world with more ease.

CHAPTER **2**

Background and Motivation

The Internet has influenced our daily lives in fundamental and profound ways. Being able to stay in touch with family and friends via email or social media websites strengthens the feeling of belonging and identification with a community, even at great distances.

In today's digital age, people with an intellectual disability (ID) often have trouble partaking in these online activities, such as using email, chat, and social network websites. Not being able to access or use information technology is a major form of **social exclusion**. There is a dire need for digital communication interfaces that enable people with ID to contact one another and their environment (teachers, parents, etc.).

The Text-to-Pictograph and Pictograph-to-Text translation systems address exactly this issue. The Text-to-Pictograph translation system (see Part II) automatically augments written text with Sclera or Beta pictographs (see section 1.3.2.3), two pictograph sets that are commonly used in Flanders, and is primarily conceived to improve the *comprehension* of textual content. The Pictograph-to-Text translation system (see Part III) allows the user to insert a series of Sclera or Beta pictographs, automatically translating the image sequence into natural language text where possible, hereby facilitating the *construction* of textual content. In the following chapters, we will provide a detailed description of both tools.

The current chapter provides background information. We first discuss the origins of the pictograph translation project (section 2.1). This part is followed by a description of the target users and the evaluation methodologies used (section 2.2). The Text-to-Pictograph and Pictograph-to-Text translation engines make use of the same underlying lexical-semantic database for Dutch. We provide a detailed description of this resource, and demonstrate how the pictographs are connected to lexical entries in the database (section 2.3). We also show how the hierarchical and relational structure of the database aids in improving the lexical coverage of the translation engines. Finally, we present our conclusions (section 2.4).

2.1 WAI-NOT: Where it All Started

The pictograph translation systems find their origins on the WAI-NOT website,[1] a safe internet environment for Dutch-speaking people with ID. WAI-NOT is a highly accessible platform that offers news articles written in easy-to-read form, text-to-speech functionalities, and educational games. Users are encouraged to build their own social network, consisting of family members and classmates, and to exchange emails and chat messages with each other using the accessible email client. WAI-NOT takes the vulnerability of its users into account by prioritising their privacy and safety, and its chat box is strictly moderated by volunteers. The website is widely spread over special needs education schools in Flanders.

The users of the WAI-NOT communication tools have two input modes, which they can combine. They can select pictographs from the Sclera or Beta pictograph set by means of a hierarchical system, or they can type text.

Figure 2.1: Erroneous translation by WAI-NOT's Text-to-Pictograph translation engine of *ik kom* 'I come' into *ik.png kom.png*, with *kom.png* representing the noun *kom* 'bowl'.

Originally, WAI-NOT made use of a very basic Text-to-Pictograph translation engine that did not involve language technology. For every word that was inserted by the user in the chat box or email client, the original translation system would attempt to match it against a list of pictograph file names. If it was able to find an exact match between the word and a file name (without the *.png* extension), the pictograph would

[1]http://wai-not.be/

be displayed. Inflected forms could not be translated at all, and erroneous translations would often arise as a result of homonymy (see Figure 2.1).

2.2 Defining the Target Users

The pictograph translation technologies that are presented in the following chapters were developed in collaboration with the target users and their direct environment (parents, teachers, etc.).

In section 2.2.1, we define the concept of *intellectual disability*. We discuss the distinction between *intellectual functioning* and *adaptive behaviour* and show that, for many people with ID, specific limitations in adaptive behaviour may lead to significant problems with respect to the use of technology, including social media websites and email. The e-exclusion issue is explained in section 2.2.2. Section 2.2.3 puts forward four personas to represent the behaviour of our hypothesised target group - we will use these personas in the following chapters to demonstrate the impact of our technologies on the daily lives of people with ID. Finally, in section 2.2.4, we discuss how user feedback and a corpus of user-generated content are used in the further development and evaluation of the tools.

2.2.1 Intellectual Functioning and Adaptive Behaviour

The American Association on Intellectual and Developmental Disabilities (AAIDD) defines an *intellectual disability* as "a disability characterised by significant limitations in both intellectual functioning and in adaptive behaviour, which covers many everyday social and practical skills".[2] **Intellectual functioning**, on the one hand, refers to general mental capacity, including learning, reasoning, and so on. It can be measured by means of an IQ test. **Adaptive behaviour**, on the other hand, encompasses the whole range of conceptual, social, and practical skills that are learned and performed by people in their everyday lives.

The end users of our pictograph translation technologies experience specific limitations in adaptive behaviour. However, these issues are not necessarily related to limitations in their intellectual functioning. A person with a mild intellectual disability may possess close to normal intelligence and is sometimes able to read or write

[2]https://aaidd.org/intellectual-disability/definition

fairly simple texts (Nomura et al. 2010:7). Those with a profound intellectual disability cannot read or write by themselves. Therefore, in the following chapters, we will use the term *intellectual disability* to refer to *adaptive behavioural challenges* with respect to the following domains:

- **Conceptual skills**: Language and literacy, money, time, and number concepts.
- **Social skills**: Interpersonal skills, social responsibility, self-esteem, social problem solving, etc.
- **Practical skills**: All sorts of activities of daily living and personal care, occupational skills, travel, safety, etc.

Applying this to the e-exclusion case, we are able to make the following observations:

- Our target users experience difficulties in use of language. They have limited reading and/or writing skills.
- Since our target users have limited access to online social environments, such as Facebook, Twitter, or email clients, their social skills can't fully flourish.
- Our target users do not often use communication devices, such as tablet computers, either because these devices are not always cognitively accessible, or because their environment (wrongly) assumes that these communication technologies are too complex for them. Nevertheless, having access to a smartphone or tablet may contribute positively to the domains of leisure and occupational skills (playing games, reading the news, watching videos, visiting fan pages, etc.), independent travel (reading travel instructions, timetables, etc.), and overall safety (being able to call or message someone, being tracked by GPS in emergency situations, etc.).

Note that the term *cognitive disability* is a broader concept, encompassing various intellectual or cognitive deficits, including, but not limited to, intellectual disabilities. A person with an extremely high IQ may still have a severe cognitive disability. For example, he/she may have a specific learning disability (such as dyslexia or dyscalculia) or have acquired a brain injury, but not have an intellectual disability.

Different needs and limitations require different, tailor-made solutions. In the following chapters, we will underline the importance of *personalisation*. A large variety of skills and limitations will be taken into account during the development of the tools. We will create optional modules and settings that can be fully customised to any user's specific needs.

2.2.2 Problem Definition

The Convention on the Rights of Persons with Disabilities, adopted by the United Nations in 2006, states that "a disability results from the interaction between persons with impairments and attitudinal and environmental barriers that hinder their full and effective participation in society". A growing body of research suggests that creating meaningful ways to access social media websites for people with ID is a helpful way to promote meaningful inclusion in society, and a fundamental right of people with a disability.

All too often, people with ID have limited social networks in their lives (Davies et al. 2015:30). With the growing role of Facebook,[3] Twitter,[4] and Instagram[5] in shaping access to social capital by facilitating direct communication, i.e., through commenting and messaging (Wilson et al. 2012:209), people with ID are at risk of exclusion.

Tuset et al. (2010:797) remark that the exclusion of people with disabilities is, in essence, twofold. Firstly, disabled people suffer from inner communication limitations, depending on the type and degree of their condition. Secondly, modern electronic communication devices pose an important accessibility barrier due to the design of their user interfaces. These limitations affect self-esteem and reduce the self-sufficiency of disabled people.

We will underline the idea that the early involvement of the users is a key factor for success in the development and following impact of the proposed technological solutions (Daems et al. 2015:79). This type of inclusive research accesses and represents the views and experiences of people with ID, as suggested by Walmsley & Johnson (2003). In the following chapters, we will show that the feedback and opinions of the end users influence many technological decisions during the development of the tools.

[3] http://facebook.com/
[4] http://twitter.com/
[5] http://instagram.com/

2.2.3 Personas

Throughout this dissertation, we will use four personas to represent the goals and behaviour of our hypothesised group of users. Broadly speaking, the technologies target two groups of users: users with a disability (the target users), and their environment. A person belonging to the first group, i.e., a person with ID, may experience difficulties in understanding oral or written language (**receptive** language issues) or struggle to put his/her thoughts into words (**expressive** language issues), or have mixed receptive-expressive language issues. The users' families and living environment belong to the second group. As communication implies the presence of both a **receiver** and a **sender**, people without ID (parents, teachers, etc.) are highly likely to be involved in the use of the pictograph translation technologies: They may want to have a better understanding of the thoughts and ideas of people with ID (as a receiver) and/or have a desire to be better understood by them (as a sender).

Target user as receiver: Link is 15 years old. He has an autism spectrum disorder and an intellectual disability. Link has a social media account, but he does not use it actively, because he often finds it challenging to understand written text. Long captions or status updates are too difficult for him to read. As a result, Link feels left out. At school and at home, Link uses Sclera pictographs to make up for his receptive language issues. Sclera pictographs aid Link in following simple directions and organising information. With the Text-to-Pictograph translation tool at his disposal, Link will henceforth be able to better understand the captions, status updates, and private messages that are sent to him on social media websites.

Target user as sender: Chara is 18 years old and has a mild intellectual disability. With the help of her mother, she created a social media account. Most of her contacts are her friends from school and her family. Chara likes to scroll through her social media news feed and spends a lot of time looking at pictures and video clips that are posted by her contacts and her favourite fan pages. However, she never posts comments and hardly ever contacts her friends or family through the messenger client, because she finds it hard to express herself, and she does not want to feel exposed. Thanks to the Pictograph-to-Text translation technology, she will be able to write her own status updates and messages using a combination of (automatically corrected) text and a very large set of Beta pictographs. Being able to communicate with people with and without an intellectual disability makes Chara feel more included.

Environment as receiver: Freya has a 27-year old daughter named Vivi. Vivi has Down Syndrome and she does not talk much, often finding it difficult to express her thoughts and ideas. Vivi currently works at a sheltered workshop, a few blocks away from home. Recently, she has decided that she wants to walk home by herself, and depend less on her employer's shuttle service, especially when the weather is nice outside. Freya acknowledges Vivi's decision, but she can't help but worry that unforeseen circumstances (running out of phone credit to make a phone call, too much traffic noise, etc.) on the way home may leave her daughter confused and feeling lost. For that reason, Freya will show Vivi how to use the Pictograph-to-Text translation system, using a limited set of Beta pictographs that are related to feelings, traffic, locations, and emergencies. Freya feels relieved, knowing that Vivi will use the pictographs to communicate with her parents about her journey.

Environment as sender: Bonnie has a 24-year old brother named Clemont, who has an autism spectrum disorder and an intellectual disability. Bonnie attends a university abroad, so she cannot visit her family often. Clemont has difficulties understanding what others are saying. Due to receptive language issues, the majority of communication technologies are not cognitively accessible for him. Bonnie finds it difficult to stay in touch with her brother. That is, until she discovers the Text-to-Pictograph translation technology. After some training, Clemont learns how to access the emails that are sent by his sister independently. Since Clemont can't read well, Bonnie's emails are automatically simplified and translated into Sclera pictographs by means of the Text-to-Pictograph translation technology. Bonnie is excited that she is now able to send personal messages to her brother and tell him about her life on campus.

2.2.4 Involving the Target Users in the Evaluation of the Technologies

The involvement of the target users in the evaluation of our technologies is twofold. On the one hand, they are *directly* involved, by means of focus groups and hands-on sessions, allowing us to observe the users' behaviour (section 2.2.4.1). On the other hand, they are also *indirectly* involved, since we dispose of a large corpus of emails that have been sent by people with ID on the WAI-NOT website, which contains valuable information on the users' spelling, vocabulary, and conversation topics (section 2.2.4.2).

2.2.4.1 Focus Groups and Observations

Qualitative research plays a valuable role in informing researchers about the experiences of people with ID (Beail & Williams 2014:85). We are using **focus groups** and **observations** as a primary means to gather data. Focus groups have previously been used successfully in experiments with the target group (Nind 2009:11), providing the advantages of a group dynamic that can help build confidence, safe environments that are not intimidating, and peer support and validation (Cambridge & McCarthy 2001:477). Prolonged engagement of the users and efforts to create rapport between the participants and the researchers ensure trustworthiness of the results (Daems et al. 2015:82).

The translation systems are tested in Belgium within the framework of the European project Able to Include,[6] and with the help of the K-Point Research group of Thomas More University College.[7]

Two criteria were used to recruit adults for the project. Firstly, participants had to show a clear desire to access social media more easily or independently, without the help of a third party. Note that most of the participants did not yet have a social media account prior to the Able to Include project. Being motivated to cooperate was a second inclusion criterion. Selection did not happen based on IQ.[8]

We discuss the results of the focus groups and observations in section 3.3.3, where we analyse the shortcomings of the baseline Text-to-Pictograph translation system, and in section 8.1.4, where we evaluate the pictograph interface.

Note that, for the evaluation of the syntactic simplification module in section 5.5.4.3, we use a different methodology: We ask the target users' environment to fill in an online survey. The reasons for this are detailed in the relevant chapter.

2.2.4.2 The WAI-NOT Email Corpus

We received a corpus of several thousands of anonymised emails that were sent through different versions of the WAI-NOT website (see section 2.1). It contains a

[6] http://able-to-include.com/

[7] http://www.k-point.be/

[8] The research was approved by the Ethical Committee of the KU Leuven, file number G-2014 1080. A simplified and pictograph-supported consent form and informative flyer was handed to the users.

total of 69,636 e-mail messages sent by WAI-NOT users, from the start of the platform on the 28th of April, 2009 until the 23rd of May, 2013.[9] The emails have an average length of 7.7 words. There are two types of messages in the corpus. Some of the emails are written by literate people, such as teachers or parents. These messages often contain a much broader vocabulary and they are the most challenging to translate into pictographs. The second type of emails are those that are written by people with ID. These messages generally contain a handful of words or pictographs, show no punctuation marks, and present several spelling errors. The corpus also contains a lot of noise. These messages either contain long strings of random characters, or they are the result of (repeatedly) clicking pictographs in the pictograph input interface. In the following chapters, we use the WAI-NOT corpus to learn about the users' spelling, vocabulary, and conversation topics.

Due to privacy issues, not much is known about the WAI-NOT users and their specific limitations in adaptive behaviour. The website has been promoted in different special needs schools and organisations in Flanders, and the users' skills range from more severe reading and writing difficulties to rather good writing and reading comprehension skills. While this lack of user profiling is a disadvantage, the large amount of user-generated content can still be considered a valuable resource. It is safe to assume that experienced social media users would make less or no use of the WAI-NOT environment, as the website primarily targets users who do not have the conceptual, social, and/or practical skills necessary to make use of mainstream websites.

2.3 The Lexical-Semantic Resource

The Text-to-Pictograph and Pictograph-to-Text translation tools make use of large databases called WordNets.

WordNets (Miller 1995) are lexical-semantic collections that organise lexical items of a given language into synonym sets or *synsets* (see Figure 2.2). Synsets are abstract identifiers that are linked to lexical items that share the same meaning, such as *bunny* and *rabbit*. The synsets are organised hierarchically by means of a large variety of relations, such as the hyperonymy relation (the *rodent* synset is a hyperonym of the *rabbit* synset), the antonymy relation (the *high* synset is an antonym of the *low* synset),

[9] Note that, for the evaluation of the new pictograph interface in Chapter 8, we used a more recent version of the corpus, i.e., emails that were sent after the implementation of the new interface in February, 2017.

or causal relationships (the *fire* synset stands in a causal relationship with the *burn* synset).

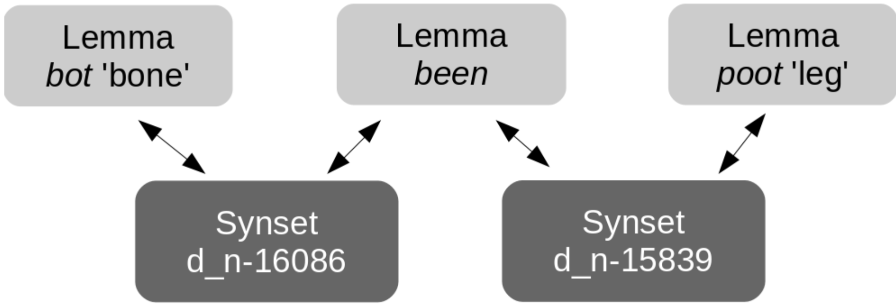


Figure 2.2: An example of a Dutch lemma, *been*, which has at least two different meanings and, therefore, belongs to at least two different synsets.

Section 2.3.1 discusses the differences between the two WordNets that are currently available for Dutch. In section 2.3.2, we explain how the gap between the Sclera and Beta pictographs and the lexical entries is bridged. Linking pictographs to WordNets is an essential step in the development of the translation technologies. Finally, we briefly discuss the potential contributions of two related lexical-semantic resources in section 2.3.3.

2.3.1 The WordNets for Dutch

Currently, there are two WordNets for Dutch available. Cornetto[10] (section 2.3.1.1) requires a license, but it contains high-quality synsets and connections. Open Dutch WordNet (section 2.3.1.2),[11] on the other hand, is an open-source resource, but is not nearly as complete or populated as the Cornetto database.

2.3.1.1 Cornetto

The Cornetto database (Vossen et al. 2008) is a lexical-semantic database for Dutch, covering over 119,000 lexical units, which are spread over more than 70,000 different synsets, including the most generic and central part of the language. In Cornetto, a lexical unit consists of an identifier and a lemma. Most synsets have a part-of-speech category encoded with them.

[10] http://wordpress.let.vupr.nl/cornetto/
[11] http://wordpress.let.vupr.nl/odwn/

Cornetto combines the structure and content of Princeton WordNet (Miller 1995) and FrameNet (Baker et al. 1998)[12] for English. Its synsets are aligned with Princeton WordNet 2.0,[13] so that ontologies and domain labels could be imported.

The first version of Cornetto was released in 2008. The database was revised during the DutchSemCor project (Vossen et al. 2012) between 2009 and 2012. The Cornetto database contains thousands of high-quality connections and its synsets are well-populated. For research, the data are licensed without a fee. For commercial usage, a fee must be paid for the background data that are delivered by the publisher Van Dale.[14]

2.3.1.2 Open Dutch WordNet

The Open Dutch WordNet was created by removing the proprietary content from Cornetto, and by replacing this content with open-source resources. The database contains 117,914 synsets, which were transferred from (and thus, are aligned with) Princeton WordNet 3.0.[15] Over 50,000 synsets include at least one Dutch lemma. Over 67,000 synsets are empty and still need to obtain a synonym. The first version of the database was released in December 2014.

Since the Open Dutch WordNet was still under development at the time of creating and improving the pictograph translation tools, and given that a large number of Dutch words or synset connections are still missing or being added over time, the tools that are described and evaluated in this dissertation make use of the Cornetto database instead, for which we obtained a license. However, once a stable and high-quality version of the Open Dutch WordNet is released, it will become possible to automatically transfer the pictograph links to the new synsets via Princeton WordNet 3.0, following the WordNet linking procedure that is described in Chapter 10.

2.3.2 Linking Sclera and Beta Pictographs to the WordNet for Dutch

In this section, we explain how the gap between the Sclera and Beta pictographs, on the one hand, and natural language words, on the other hand, is bridged. As argued

[12] FrameNet is based on a theory of meaning called Frame Semantics (Fillmore 1982). The basic idea is that the meanings of most words can best be understood on the basis of a semantic frame: a description of a type of event, relation, or entity, and the participants in it.

[13] http://wordnetcode.princeton.edu/2.0/

[14] http://www.vandale.be/

[15] http://wordnetcode.princeton.edu/3.0/

in section 2.3.2.1, the Cornetto database plays a central role in this. In section 2.3.2.2, we describe the dictionary, which contains words that are not covered by Cornetto.

2.3.2.1 Linking Pictographs to Synsets in Cornetto

The pictographs in the Dutch translation system are not linked to individual words or lemmas, but to synsets in the Cornetto database. This greatly improves the lexical coverage of the translation system, since not only one lexical surface form, but also all of its synonyms are covered by a pictograph. Additionally, if the Text-to-Pictograph translation system is not able to find a pictograph for the synset of the input word, the semantic relations between the synsets can be used to find a pictograph with a similar meaning (see section 3.1).

Vandeghinste & Schuurman (2014) manually linked a subset of 5,710 Sclera and 2,760 Beta pictographs to synsets in the Cornetto database. The annotators used a tool that allowed them to select the correct Cornetto sense of each pictograph entry. In the case of complex pictographs, such as the one depicting a verb and its object (such as *to eat a sandwich* in Sclera), the authors retrieved multiple synsets. One of these synsets was identified as the head synset, while the other synsets were marked as dependent synsets. The WAI-NOT email corpus (see section 2.2.4.2) was used to make sure that all content words that appear at least 50 times were covered by a pictograph.[16]

Vandeghinste & Schuurman (2014) observed that many commonly used Flemish variants of Dutch lexical items were missing from Cornetto, whereas the WAI-NOT website is mainly intended for the Flemish part of Belgium. Therefore, they added Flemish synonyms to the synsets from which they were lacking. Examples of this are *patat* as a synonym for *aardappel* 'potato', *camion* as a synonym for *vrachtwagen* 'truck', and *hesp* as a synonym for *ham* 'ham'.

Cornetto provides many synonyms for sexual concepts. For instance, the word *poes* 'pussycat' can be used as a synonym for the concept of female genitalia. The annotators judged the appropriateness of each lemma belonging to sexual concepts and disabled some of their senses to avoid misconversions.[17]

[16]Pictographs for frequent concepts that were missing were made on request.

[17]Actual "taboo" words, however, are still converted into their appropriate pictograph counterpart, such as the ones that depict sexual activities or genitalia. We should not forget that these words are part of the daily vocabulary of teenagers and adults with ID, or that they are used in sexual education courses. Conversely, these pictographs are included in the pictograph interface, allowing users to create

2.3.2.2 A Dictionary for Dutch Words Not Covered by Cornetto

The Cornetto database has its limitations. It contains nouns, verbs, adverbs, and adjectives, but it does not contain pronouns, interjections, or prepositions, among other things. More recent concepts, such as *smartphone* 'smartphone', or brand names, such as *PlayStation* or *iPod*, are missing. Yet, pictographs are available for these concepts.

The dictionary allows to bypass the semantic analysis via Cornetto and provides a direct link between the *token/lemma/tag* fields and the file name of a pictograph.[18] The *tag* field and either the *lemma* or *token* field can be left underspecified. The dictionary can be used to cover any words that are missing from the database.[19] For instance, a direct link was established between the interjection *hey* 'hey' and the Sclera pictograph *hallo-zeggen-2.png* 'to-say-hello-2.png'.

2.3.3 Other Lexical-Semantic Resources

BabelNet (Navigli & Ponzetto 2012) is a multilingual encyclopedic dictionary, with lexicographic and encyclopedic coverage of terms, and a semantic network which connects concepts and named entities in a very large network of semantic relations. It is automatically constructed by means of a methodology that integrates lexicographic and encyclopedic knowledge from a large variety of resources, such as Wikipedia[20] and FrameNet (Baker et al. 1998). BabelNet is made up of about 16 million entries, called Babel synsets. Each Babel synset represents a meaning and contains all the synonyms which express that meaning in a range of different languages. As of February 2018, BabelNet covers 284 languages, including all European languages. The Dutch WordNet resource that is integrated in BabelNet is Open Dutch WordNet, which was still under development at the time of creating and improving the pictograph translation tools (see section 2.3.1.2). However, there are still many reasons why using BabelNet as a core resource for image-based translation technologies could be considered useful. For instance, BabelNet includes the database ImageNet (Deng et al.

messages in the Pictograph-to-Text translation engine. However, they may not be appropriate for all users. This can easily be solved by means of a filter that blocks the generation of, or the access to, pictographs that carry a "taboo" label.

[18]Where possible, we added missing lexical units to already existing synsets.

[19]For nouns, verbs, adverbs, and adjectives, another possibility would be to create a brand-new synset, and attach a pictograph to that synset. We have chosen not to modify Cornetto's semantic network, and to use a dictionary instead.

[20]https://www.wikipedia.org

2009). In ImageNet, each node of the conceptual hierarchy is depicted by hundreds and thousands of images, which can be linked to WordNets (semi-)automatically, and which can be used to develop picture translation technologies for new target groups, such as second language learners. Note, however, that ImageNet currently only includes nouns, but no verbs, adjectives, and so on.

ConceptNet (Speer & Havasi 2012) is used to create word embeddings, i.e., representations of word meanings as vectors, which are multilingual and aligned across languages. It extends WordNet's notion of "a node in the semantic network" from purely (simplex) lexical items to include compound concepts, such as *grocery store* or *to buy food*. Whereas similar large-scale semantic knowledge bases, like WordNet, are largely handcrafted, ConceptNet is generated automatically, and its emphasis on informal, conceptual connectedness over formal, linguistic rigor allows it to go beyond WordNet to make practical, context-oriented, and commonsense inferences over real-world texts.[21] While integrating ConceptNet as a lexical-semantic resource is beyond the scope of this dissertation (which builds on earlier work that revolves around Cornetto), the potential added value of using automatically generated word embeddings over WordNet synsets within the context of pictograph translation is a topic worth investigating in future research.

2.4 Conclusion: Background and Motivation

In this chapter, we described the origins, the end users, and the lexical-semantic resource of the Text-to-Pictograph (Part II) and Pictograph-to-Text (Part III) translation tools. By making use of WordNets, such as the Cornetto database for Dutch, the pictographs can be connected to synsets and their semantically related concepts. Not only does this method improve the coverage of our tools, using the WordNet approach also allows us to transfer the pictographs to other natural languages. The main reason for this is that WordNets of different languages are connected to each other by means of their synsets, as will be shown in Chapter 10. In the following chapters, we will show how the connections between synsets and pictographs can be used to translate natural language text into pictographs, and vice versa.

[21] http://www.kmworld.com/Articles/News/News-Analysis/Where-Is-ConceptNet-Watson-92666.aspx

Part II

Text-to-Pictograph Translation

Outline

The Text-to-Pictograph translation tool automatically translates natural language text into Sclera and Beta pictographs. The baseline version of the Dutch translation engine was developed by Vandeghinste et al. (2017) for the WAI-NOT environment (see section 2.1), with the objective of facilitating online communication and social media involvement for people with an intellectual disability. The pictographs are not meant to replace written text. Rather, they can be used as a stepping stone toward a better *comprehension* of written content.

Text-to-Pictograph translation also presents a number of advantages for the target users' environment, i.e., family members, caregivers, and teachers. Browsing large pictograph databases to find the appropriate images for the construction of pictograph-supported instructions, schedules, and menus can be a tedious job. By automatically translating a written message into a series of pictographs, the Text-to-Pictograph translation engine allows family members and caregivers to send pictograph-based emails to the target group, making it simpler for them to communicate in an online setting, where the use of written text would normally cause big difficulties.

In this part, we first describe the architecture of the baseline translation system (Chapter 3). We discuss the shortcomings of the tool, and propose and evaluate three major improvements. The first improvement is a spelling correction tool for text written by people with ID (Chapter 4), as erroneous input often leads to incomplete or incorrect pictograph translations. The second solution comes in the form of deep linguistic analysis (Chapter 5), which allows the tool to syntactically analyse the input sentences before converting them into simplified pictograph sequences. Finally, we describe the implementation of a word sense disambiguation module (Chapter 6), which aids in determining the appropriate sense, and thus the appropriate pictograph translation, of ambiguous word forms. To conclude, we evaluate the improved Text-to-Pictograph translation system pipeline (Chapter 7).

CHAPTER 3

The Baseline System

The Text-to-Pictograph translation tool consists of various sub-processes.[1] We first describe the architecture of the baseline system (section 3.1), which was developed by Vandeghinste et al. (2017). A running example is given for the input sentence shown in Figure 3.1. The baseline system contains a number of parameters, which allow the engine to calculate an optimal translation. Before testing the translation engine, we had to optimise these parameters through an automated procedure (section 3.2). We then evaluate the baseline system (section 3.3) and describe its shortcomings (section 3.4).

Figure 3.1: An example sentence translated into Sclera pictographs.

[1]Previous versions of sections 3.1 to 3.3 appeared in Vandeghinste et al. (2017) and Sevens et al. (2015a).

3.1 Architecture

Input text in the Text-to-Pictograph translation tool first undergoes **shallow linguistic analysis**. The *tokenisation* module splits all the punctuation signs from the words, with the exception of the hyphen and the apostrophe. The next sub-process is *basic spelling correction*. A detailed description of this module is given in Chapter 4. *Part-of-speech* tagging is applied, using the open-source tagger HunPos (Halácsy et al. 2007). HunPos was trained on two corpora: Corpus Gesproken Nederlands (more specifically, CGN Core) (Oostdijk et al. 2002) and the Lassy-small corpus of written Dutch (van Noord et al. 2013). Both resources contain about one million words each with manually corrected part-of-speech tags.[2] Since the Text-to-Pictograph translation engine works on the sentence level, automated *sentence detection* is applied, by means of a set of handcrafted rules based on punctuation signs. The system also detects *separable verbs*, verbs that have a lexical core and a separable particle, a phenomenon particular to Dutch (and German, Afrikaans, and Hungarian). For each of the words of a sentence that is tagged as a verb, the system combines it with each of the words that is a potential particle. The module checks if the compound verb is more likely than its parts kept separately, following the method of Vandeghinste (2002). Finally, *lemmatisation* is applied by looking up the token and part-of-speech tag combination in the already mentioned corpora Corpus Gesproken Nederlands and Lassy-small.

Next, the **semantic analysis** module is activated (see Figure 3.2). The system first looks for words with a negative polarity, such as *geen* 'no' or *niet* 'not'. When such a word is found, it looks for its head and adds the value *negative* to its polarity feature. For every input word in the source message, the system returns all possible WordNet synsets for that word (see section 2.3). Synsets are encoded with a part-of-speech category, which allows the system to keep only those synsets where the part-of-speech category agrees with that of the input word. Note that this system does not yet apply proper word sense disambiguation, but uses the most common sense, based on DutchSemCor (Vossen et al. 2012). For every synset that has been retrieved, the system checks whether a connected pictograph can be found (see Figure 3.3). This pictograph connection does not necessarily have to be a direct one. Pictographs can also be retrieved via *synset propagation*, i.e., by making use of Cornetto's relations between synsets. The following semantic relations are used:

[2]The tags consist of a prefix in uppercase which indicates the main word class, and a suffix in round brackets indicating the value of the features of this word class. See Van Eynde (2005) for more details.

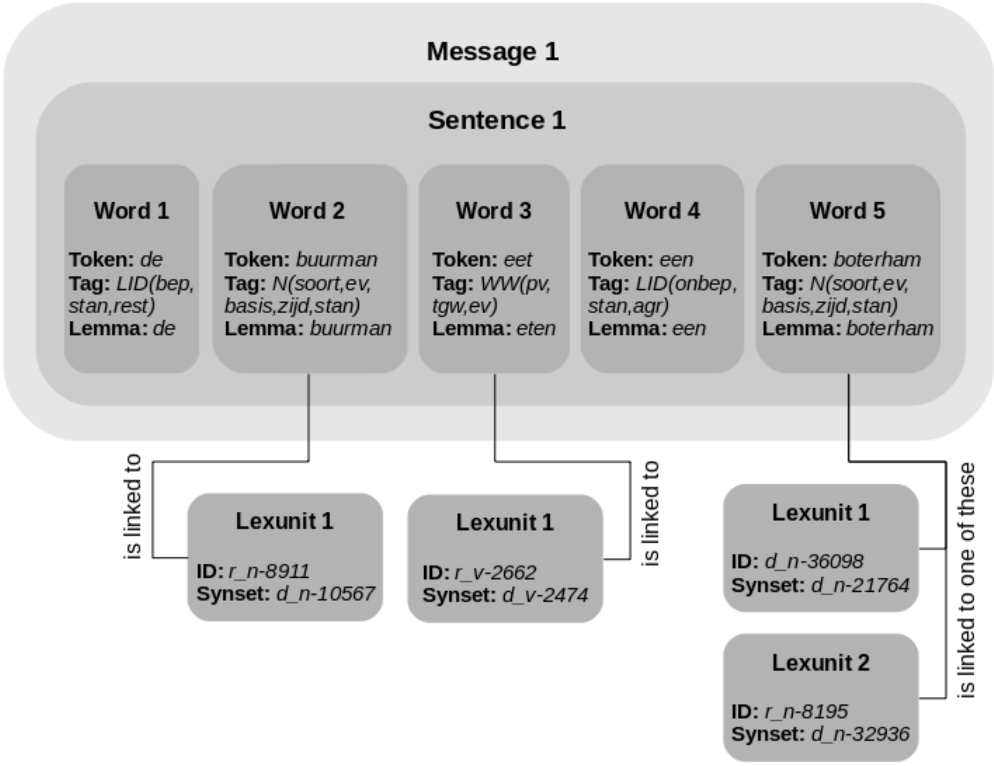


Figure 3.2: The example sentence after semantic analysis.

- **HAS_HYPERONYM**: The *hyperonymy* relation indicates the link between a subcategory and a supercategory. For instance, the word *zalm* 'salmon', for which no specific pictograph is available, is connected through its synset with its hyperonym synset *vis* 'fish' by means of the *HAS_HYPERONYM* relation.

- **XPOS_NEAR_SYNONYM**: Another relation which allows to improve the coverage of the system is the *xPos near synonym* relation. It indicates a link between similar concepts, but with a different part-of-speech tag. For instance, if no pictograph is found for the noun *gevoel* 'feeling', the *XPOS_NEAR_SYNONYM* relation can be used to retrieve the pictograph that is connected to the synset of the verb *voelen* 'to feel'.

- **ANTONYM**: The final relation that the system uses is the *ANTONYM* relation, indicating that synsets are the opposite of each other. Since no pictograph is available for *genezen* 'recovered', its antonym *ziek* 'ill' is retrieved, along with a negation pictograph, resulting in the sequence *niet ziek* 'not ill'.

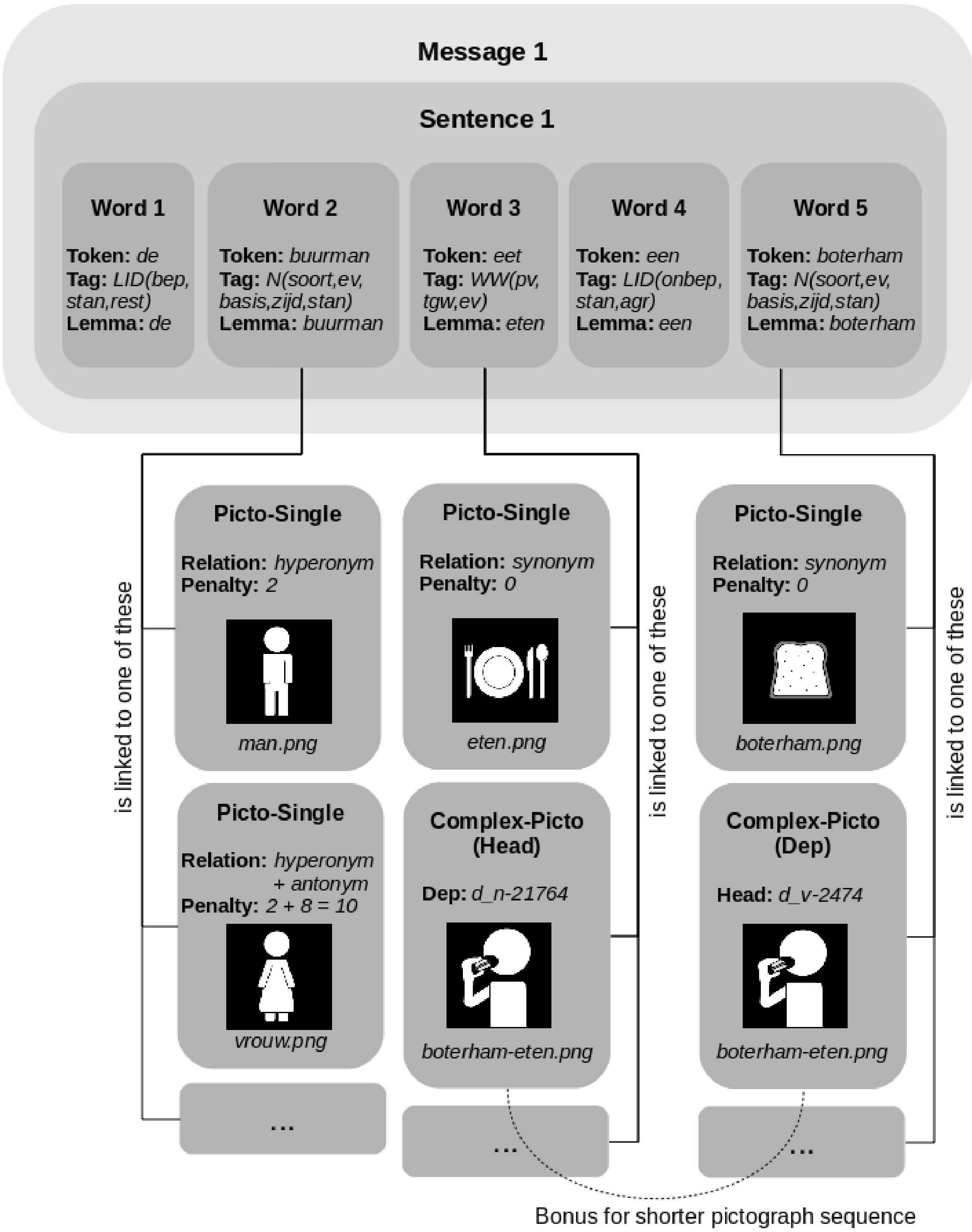


Figure 3.3: The example sentence with linked Sclera pictographs. There exists no pictograph for *buurman* '(male) neighbour', but the translation engine manages to retrieve its hyperonym *man* 'man', and a number of synsets that are connected to that hyperonym, such as the antonym *vrouw* 'woman'. At least two valid translations for *eet een boterham* 'eats a sandwich' are found, but the use of a complex pictograph is preferred over the use of two separate pictographs, as it shortens the output sequence.

To deal with pronouns and words that have a part-of-speech tag that is not covered by Cornetto, the **direct route** is introduced. It provides direct mappings between pronouns and their corresponding pictographs, using the person, gender, and number information that was obtained during the part-of-speech tagging process, and makes use of the handcrafted pictograph dictionary (see section 2.3.2.2).

For every word in the sentence, the system checks whether one or more pictographs can be found for it and whether the use of these pictographs is subject to a penalty. For instance, using related concepts through synset propagation is penalised. Penalty weights correspond to **parameters** (see section 3.2).

Finally, an A* algorithm calculates the **optimal pictograph sequence** for the source text. Its input is the pictographically annotated source message, together with the pictographs' penalties, depending on the number and kind of synset relations the system had to go through in order to connect them to the input words. The algorithm starts with a queue containing an empty path that still has all the input words left to process. In every step, the currently best scoring pictograph path is extended: The system checks whether there are any pictographs, along with their corresponding penalties, that are connected to the next word to be processed. New paths are created by adding the retrieved pictograph to the list of the already matched pictographs. All possible paths are added to the queue. The queue is sorted by lowest estimated cost and the best scoring path is extended.[3] This process is repeated until the first queue element no longer has any words left to process. When encountering words that have their antonym feature set to *negative*, the negation pictograph is inserted.[4]

3.2 Parameter Tuning

In order to evaluate the baseline version and to analyse its shortcomings, we must build the best possible version of the translation engine, i.e., by tuning the system's parameters. Section 3.2.1 describes the parameters that are used in the baseline system. Section 3.2.2 explains the parameter tuning process.

[3]The use of complex pictographs, which merge multiple concepts within one pictograph, is preferred by the system over the separation of those concepts. The shorter the pictograph translation, the higher it will be scored by the system.

[4]Vandeghinste et al. (2017) provide a detailed description of this algorithm.

3.2.1 Description of the Parameters

The first set of parameters (*hyperonym penalty*, *xPos penalty*, and *antonym penalty*) concerns the maximum distance (*threshold* parameter) allowed between the original text and the pictograph message in terms of synset relations.

The second set of parameters is related to the numeric features of the pictographs (*no number* and *wrong number*), as some pictographs make a distinction between singular or plural concepts (such as *oog.png*, depicting one eye, and *ogen.png*, depicting two eyes).[5]

3.2.2 Tuning the Parameters

The parameters are tuned using a set of 50 sentences (653 words) from the WAI-NOT corpus (see section 2.2.4.2 and Appendix B.1), which we manually translate, to the best of our ability, into Sclera and Beta pictographs. These manual translations form the reference set.

Parameter	Min	Max	Step	Sclera	Beta
Cornetto relations					
Threshold	5	20	1	20	17
Hyperonym penalty	0	15	1	2	8
XPos penalty	0	15	1	7	7
Antonym penalty	0	15	1	8	7
Pictograph features					
Wrong number	0	10	1	8	8
No number	0	10	1	6	5

Table 3.1: Parameter values for the baseline Text-to-Pictograph translation system after tuning.

We tune the parameters by means of an automated procedure, using a local hill climber (Vandeghinste et al. 2017) that varies the parameters when running the Text-to-Pictograph translation script on the two test sets (from Dutch to Sclera and from Dutch to Beta). The BLEU metric (Papineni et al. 2002) is used as an indicator of relative improvement. BLEU is a precision-oriented metric which compares the system

[5]Since pictographs with numeric features do not occur often, we verified their presence in the tuning set. We were able to find a number of occurrences (such as *vrienden* 'friends' and *luizen* 'lice').

output to one or more reference translations, by counting how many *n*-grams overlap, and correcting for brevity.

In order to find the highest BLEU score, we run five trials of the local hill climbing algorithm for both the Sclera and the Beta condition, until BLEU converges onto a fixed score, after several thousands of iterations. We run each trial with random initialisation values for the parameters, while varying them between certain boundaries, and with a parameter step size of one, in order to cover different areas of the search space. From these trials, we choose the best scoring parameter values for the Sclera and the Beta condition. These values are presented in Table 3.1.[6]

3.3 Evaluation of the Baseline System

Having obtained the optimal parameter values, we can now move on to the evaluation of the baseline system. Section 3.3.1 shows the procedures we apply for automated evaluation. In section 3.3.2, we describe a manual evaluation, comparing the results of translating into Sclera with the results of translating into Beta. Finally, we present the opinions and experiences of the target users in section 3.3.3.

3.3.1 Automated Evaluation

For the evaluation of the baseline system, we select another 50 Dutch messages (84 sentences or 980 words) that were sent with the WAI-NOT email system (see section 2.2.4.2 and Appendix B.2). Our test set consists of a combination of messages written by literate and illiterate users. Noisy messages are not included. For each of these sentences, we create one reference translation in Sclera and one in Beta, translating, to the best of our abilities, the messages into the respective pictograph sets. We focus on the content of the message and how this content can most clearly be expressed in pictographs. We do not post-edit the system's output.

We automatically evaluate different experimental conditions, progressively activating more modules of the system. The first condition is the *WAI-NOT* baseline condition, in which the system only replaces words by pictographs if the pictographs have the same file name (without the *.png* extension) as the input words (see section 2.1). In the next condition, we apply *lemmatisation*. The input sentences are tokenised,

[6] With Sclera's threshold value hitting the upper boundary, it could be worth repeating this experiment with wider limits. This is future work.

part-of-speech tagged and lemmatised, and words that have a lemma that has the same file name as the pictograph are translated. We then add the *direct route*. This includes a specific treatment of the pronouns, and a set of dictionary entries. The following condition uses the *synsets* of the Cornetto concepts: The pictographs are linked to synsets, and the input words are also linked to synsets. Whenever they share a synset, the words are translated into pictographs. The last condition also uses the *relations* between synsets.

Table 3.2 shows the respective **BLEU**, **NIST** (Doddington 2002), **Word Error Rate** (WER) and **Position-independent word Error Rate** (PER) scores for the translation of messages into Sclera and into Beta. NIST is similar to BLEU, but gives less credit to high frequency non-informative n-grams. WER is often used in speech recognition and counts the number of words that are incorrect with respect to the reference translation(s). PER is like WER, but treats the words as a bag. We include PER as there is no language model available for Sclera or Beta. In other words, the position of the pictographs is not necessarily what we want to evaluate.

For each condition, we include three variants. *No spelling correction* takes the input as such and excludes the basic spelling correction process that is part of the shallow linguistic analysis step. *Automated spelling correction* takes the input as such and applies the basic spelling correction process before translating the text into pictographs. Finally, in the *manual spelling correction* variant, we have manually corrected the spelling of the input, to the best of our abilities, and sent this input to the translation engine. This is done to calculate an upper bound, in order to estimate how much more room for improvement there still is in the spelling correction process.

We add significance levels for the BLEU and NIST scores, by comparing each condition with the condition on the previous line. Significance is calculated using bootstrap resampling (Koehn 2004).[7]

Table 3.2 proves that the Text-to-Pictograph translation tool improves translation quality over the WAI-NOT system. The effect of the spelling corrector is negative or very small (and insignificant for BLEU) in the initial conditions, but it becomes significant for BLEU, as well as NIST, in the more advanced conditions (for instance, *synonyms* and *relations*).

[7]Bootstrap resampling is one of the earliest randomised methods proposed for statistical significance testing of machine translation systems, to assess for a pair of systems how likely a difference in BLEU scores occurred by chance (Graham et al. 2014:266).

	No spelling correction				Automated spelling correction				Manual spelling correction			
Condition	**BLEU↑**	**NIST↑**	**WER↓**	**PER↓**	**BLEU↑**	**NIST↑**	**WER↓**	**PER↓**	**BLEU↑**	**NIST↑**	**WER↓**	**PER↓**
Sclera												
WAI-NOT	00.00	1.43	96.27	92.13	00.00	1.38	99.00	95.58	00.00	1.84	97.51	94.20
Lemmatis.	01.87*	1.68**	94.48	89.36	01.91*	1.73**	93.92	88.81	02.44*	2.29**	92.27	87.02
Direct	10.74**	2.93**	75.41	69.20	11.57**	3.05**	74.59	68.09	14.17**	3.68**	71.96	65.88
Synonyms	**12.02***	**3.32****	**70.58**	**63.26**	**13.24***	3.41**	**70.03**	**62.43**	**16.55****	*3.97***	*67.54*	*60.50*
Relations	11.44	3.29	72.24	64.50	12.75	**3.42**	71.41	63.26	16.12	3.96	68.78	61.33
Beta												
WAI-NOT	05.93	2.29	80.76	72.21	04.94	2.40	81.10	71.99	04.70	2.81	79.19	69.85
Lemmatis.	07.77	2.90**	77.05	66.93	08.15	3.01**	76.94	66.14	10.14**	3.53**	74.92	63.78
Direct	11.96**	3.65**	66.59	57.48	12.72**	3.76**	66.14	56.47	16.98**	4.43**	63.44	53.77
Synonyms	16.57**	4.12**	**56.24**	**46.91**	18.70**	4.28**	**55.12**	**46.01**	23.01**	5.00**	52.42	43.31
Relations	**18.56***	**4.22****	56.47	47.24	**20.11***	**4.40****	55.46	**46.01**	**25.91****	**5.17****	**51.29**	**42.07**

$^{*}p < 0.05, ^{**}p < 0.01$.

Table 3.2: Automated evaluation of Text-to-Pictograph translation with one reference translation for the different experimental conditions.

Adding *lemmatisation* results in a substantial improvement. For Sclera, there is significant improvement in comparing the *WAI-NOT* baseline condition with *lemmatisation* in all three variants, both for BLEU and NIST. In Beta, the rise in BLEU and NIST is smaller (because the *WAI-NOT* baseline has a better score), but still significant for NIST in all three variants, and significant for BLEU in the case of manual spelling correction.

Adding the *direct route* leads to a clear and significant improvement for Sclera and Beta in both BLEU and NIST in all three variants. This can be attributed to the proper treatment of high frequency phenomena. Adding the *synonyms* results in another significant improvement for both target languages in BLEU and NIST in all three variants, as this step greatly improves the coverage of the system. The effect of adding the *relations* is less straightforward. It does not seem to improve scores in translations into Sclera, in any of the three variants in any of the metrics, although the decrease is not statistically significant either. For Beta, we see that there is a significant improvement for BLEU and NIST when the relations are added to the system, for all three variants.

There is quite a large gap between the results for Sclera and the results for Beta. To find an explanation for this gap, we performed more experiments. The Sclera pictograph set consists of a much larger amount of pictographs than Beta, and it is, therefore, more difficult to manually translate text into Sclera messages. Several different paraphrasing translations are possible, resulting in a less accurate measurement of translation quality.

To test this hypothesis, we have created a second reference translation for Sclera, based upon post-editing the system output. Figure 3.4 shows the effect this has on the BLEU score, as compared to the evaluation with the manually created reference, as presented in Table 3.2. For the post-edited reference, the effect of adding the synsets is significant in all three variants. The effect of adding the relations, however, remains insignificant for NIST and BLEU for all variants, apart from NIST on the automated spelling correction variant.

While Vandeghinste et al. (2017) make significant improvements with their baseline translation system, as compared to the WAI-NOT baseline system, there is still room for improvement. For instance, the scores in the *automated spelling correction* condition are significantly lower than those in the *manual spelling correction* condition. This finding suggests that more appropriate correction methods, preferably tailored toward the characteristics of the target group, are necessary.

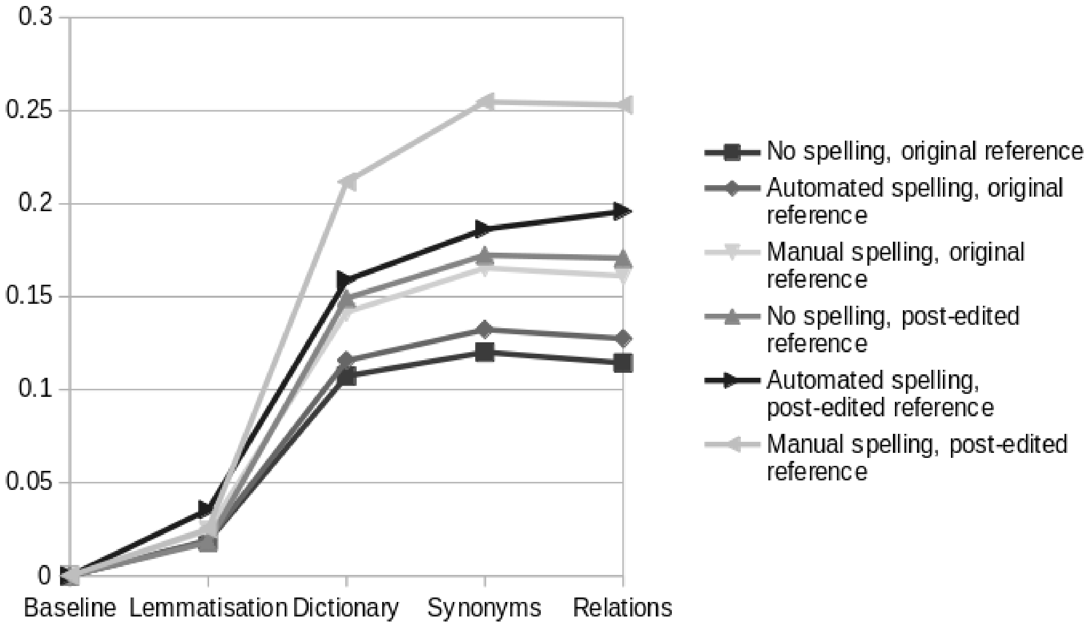


Figure 3.4: BLEU scores for Sclera for all experimental conditions in the three variants, with post-edited reference translations.

3.3.2 Manual Evaluation

We also perform a manual annotation, removing untranslated words that are considered not to contribute to the content. This allows calculating the **recall**. For each of the translated words, we judge whether the pictograph generated is the correct pictograph, in order to calculate **precision**. Recall and precision are combined in an **F-score** (van Rijsbergen 1975). The results are presented in Table 3.3.

As proper names occur frequently in email messages, we calculate recall and F-score with and without proper names, in the latter case removing all proper names from the output. Precision remains the same in both conditions. In the case where proper names are included, they are not converted into pictographs, affecting recall negatively. In the WAI-NOT environment, proper names occurring in the contact lists of the users are converted into the pictures attached to these profiles, resulting in more personalised messages.

The improvements for Sclera are very large, especially the rise in recall, although we still have a substantial rise in precision as well. The Beta WAI-NOT baseline system was already much better than the Sclera WAI-NOT baseline system, so the improvements for Beta cannot be of the same magnitude. Nevertheless, there is still a

		Without proper names		With proper names	
Condition	**Precision**	**Recall**	**F-score**	**Recall**	**F-score**
Sclera					
WAI-NOT	77.60%	41.42%	54.01%	36.39%	49.55%
Text-to-Picto	89.24%	86.23%	87.71%	85.18%	87.16%
Rel. improv.	*15.00%*	*108.19%*	*62.39%*	*134.06%*	*75.92%*
Beta					
WAI-NOT	82.73%	62.23%	71.03%	59.57%	69.27%
Text-to-Picto	85.91%	89.45%	87.64%	88.68%	87.27%
Rel.improv.	*3.84%*	*43.73%*	*23.38%*	*48.88%*	*26.00%*

Table 3.3: Manual evaluation of the Text-to-Pictograph translation engine.

substantial rise in recall, and a small rise in precision.

The F-scores for both target pictograph sets, while very different in the baseline, now lie around 87%. The difference in performance of the WAI-NOT baseline systems led to the fact that Sclera was nearly unusable. This is now resolved, with both the translation of text into Beta and and the translation of text into Sclera reaching similar levels of accuracy. However, there is still room for improvement.

3.3.3 User Studies

We tested the baseline Text-to-Pictograph translation system with a focus group (see section 2.2.4.1) of five adults (one man and four women) with ID and two coaches in a day centre in Flanders. The system was evaluated on Facebook and Facebook Messenger, using tablet computers.[8] During the hands-on session, participants were prompted to use the Text-to-Pictograph translation tool on their news feed to translate questions, captions, and comments. Additionally, private chat messages were exchanged between participants and their coaches on Facebook Messenger, and translated into pictographs.

We prepared a number of questions, which we used as a guideline for our conversations with the target users and their coaches during the hands-on session:

[8]Within the Able to Include project (see section 2.2.4.1), the "Able Social" app was developed, allowing people with ID to use the pictograph translation technologies (and other assistive technologies, such as a text-to-speech tool) in social media environments, such as Facebook and Twitter. A developer's kit of the Able Social app is available on Github: https://github.com/able-to-include/ABLE-Social-Media-Accessibility-App

- Do you like to use the Text-to-Pictograph translation engine?
- Would you like to continue using social media websites?[9]
- Are text-to-speech solutions more useful than pictographs?
- What makes the Text-to-Pictograph translation engine difficult to use? Did you encounter any weird translations?

The participants enjoyed using the Text-to-Pictograph translation technology, claiming that the tool was easy to use. During the hands-on session, one of the users found a picture of her coach, wearing funny clothes. She used the pictograph technology to read the caption of the picture. The pictographs revealed that the coach went to a Halloween party. The user and her friends started to laugh, and discussed where and when the Halloween party took place.

One of the coaches underlined the importance for the participants to learn how to use social media independently. She noticed that Facebook aided them in strengthening the emotional bond with their family members, especially family members who do not visit them often because they live too far away, or helped them stay in touch with their friends.

While all five participants knew how to use Skype[10] and text-to-speech solutions, they imagined a number of situations in which they would choose pictographs, rather than opt for video chat or read-aloud text. Privacy (i.e., making sure that their private conversations are not overheard by others) was the most important factor. Furthermore, they remarked that there is often too much noise to be able to hear the output of text-to-speech technologies, especially in public spaces. In these situations, pictograph translations would be helpful, they claimed.

The focus group also revealed a number of obstacles. Firstly, even when being able to read and write, most of the participants still struggle to spell correctly (Daems et al. 2015:78). When erroneous input is fed into the Text-to-Pictograph translation engine, the baseline system cannot always provide a (correct) pictograph translation. Yet, spelling errors occur frequently on social media websites - especially when the user's network mainly consists of people with writing difficulties.

[9]Note that privacy and rules of conduct were discussed in training sessions prior to the Text-to-Pictograph evaluation session.

[10]https://www.skype.com/

Secondly, participants stumbled upon long and complex input sentences, such as some of the captions that are written by the administrators of their favourite fan pages, resulting in long and complex pictograph translations. The baseline translation system provides an almost literal pictograph translation and it does not perform deep linguistic analysis. During the hands-on session, long and complex pictograph sequences led participants to experience more difficulties in understanding the meaning of these messages, as their screens would be completely flooded with pictographs. Another reason why literal pictograph translations are difficult to read is the fact that the Sclera and Beta language lack many grammatical properties, such as tense or aspect, and the fact that there are hardly any pictographs for function words.

In conclusion, while the baseline tool was deemed useful and necessary by the target users, the pictograph translations were often not considered accurate or simple enough to make the engine truly usable for them.

3.4 Shortcomings of the Baseline System

The baseline translation system certainly has its merits, but our evaluations show that certain improvements can be made.

There is clearly room for further improvement in the **automated spelling correction process**, as the scores for the upper bound in the advanced conditions are significantly better than the scores for the automated spelling correction process. Untranslatable input affects recall in the Text-to-Pictograph translation system negatively. More importantly, the Text-to-Pictograph translation engine is designed to be used in online social environments, where spelling errors are omnipresent. If too much input text cannot be translated into pictographs, the target users might abandon the translation technology. Therefore, we propose a new method toward spelling correction in Chapter 4.

Secondly, linguistic analysis in the baseline system is shallow, and it is mostly limited to the word level. Complex input results in complex output, leaving the end users confused and distracted. By adding deep linguistic analysis to the translation pipeline, syntactic simplification rules can be applied, resulting in shorter, consistent, and more readable pictograph translations. Our approach toward **syntactic simplification** for pictograph translation is described in Chapter 5.

Finally, the baseline Text-to-Pictograph translation system does not yet perform semantic analysis to select the appropriate sense of a word before converting it into a pictograph. Instead, it picks the most frequent sense. Erroneous pictograph translations affect precision in the Text-to-Pictograph translation system negatively. For that reason, we introduce **word sense disambiguation** in the translation process. This procedure is described in Chapter 6.

Each of these improvements will be presented as an optional module that can be enabled or disabled by a parent or a caregiver.

Persona: Link (Before)
Link has reading difficulties. While he loves browsing through photo albums and watching videos on social media websites, he very often feels like he is missing out on a lot of information, due to his limited understanding of written text. The Text-to-Pictograph translation tool has proven to be quite a useful tool for him - especially when the input sentences are short -, but in most cases, the translations are still too complex for Link to get a real grasp of what is going on. Even more, when people use chat language, many words cannot be translated, or erroneous translations are generated.

CHAPTER 4

Improvement #1: Automated Spelling Correction for People with ID

In the pre-processing phase of the baseline Text-to-Pictograph translation system, basic spelling correction is applied, as some users are able to write short messages without relying on pictograph input (see Chapter 8). While it is important to encourage people with ID to write their own messages, independently, if they have the ability to do so, their writings are often riddled with spelling mistakes. This has several consequences. Firstly, even if the receivers of the ill-formed messages are, to some extent, able to read written text, they might still not be able to understand these messages because there are simply too many mistakes. Secondly, as noted by Sproat et al. (2001), text normalisation is recommended before applying a more conventional natural language processing technique. The Text-to-Pictograph translation tool may retrieve wrong pictographs or no pictographs at all when encountering erroneously written words.

With no freely available tools for automated Dutch spelling correction at their disposal, Vandeghinste et al. (2017) created a simple, handcrafted spelling corrector for the baseline translation system. It uses the open-source lexicon from OpenTaal,[1] which is the Dutch lexicon used in freely available spell checkers.[2] Additionally, in order to avoid the correction of anthroponyms, a list of first names is consulted.[3] For

[1]www.opentaal.org

[2]Note that a spell checker is not the same as a spelling corrector. A spell checker proposes solutions for words that are not stored in its dictionary and requires the user to choose a correct variant.

[3]The original list is no longer available, but an updated list can be found at https://statbel.fgov.be/en/male-and-female-first-names

every word that is not found in the lexicon or the list of first names, a number of real-word variants are generated by applying one character deletion, one insertion, or one substitution. The winning alternative, if any, is the one that has the highest frequency.[4]

In the previous chapter, we evaluated the baseline Text-to-Pictograph translation system (see section 3.3) and showed that there is clearly room for further improvement in the automated spelling correction process, as the scores for the upper bound (manual spelling correction) are significantly better than the scores for the basic, automated spelling correction process. An example of erroneous Text-to-Pictograph translation is shown in Figure 4.1.

Figure 4.1: Example of erroneous Text-to-Sclera translation.

We develop an automated spelling corrector that is specifically tailored to users with ID.[5] After a discussion of related work (section 4.1), we present a comparison between messages that were sent with the WAI-NOT system and tweets, and show that users with ID make *more* and *different types* of spelling mistakes than users who do not have ID (section 4.2). Next, we describe the system architecture (section 4.3). The correction process starts off with a variant generation and filtering step that is largely based on automatically discovering phonetic similarities. We then apply character-based fuzzy matching to choose the correct variants. Our evaluations show that improvements over the baseline in the Text-to-Pictograph translation tool are made (section 4.4). Finally, we conclude and describe future work (section 4.5).

[4]The frequency list was created by Vandeghinste (2002) and contains roughly eighty million tokens worth of Belgian newspaper text.

[5]Previous versions of sections 4.1 to 4.5 appeared in Sevens et al. (2016b).

Persona: Chara (Before)
Chara loves to spend time on Facebook, but she hardly ever posts status updates or comments, because spelling is difficult for her. She is scared that people will make fun of her, especially in fan communities, where most people do not know about her disability. Even her classmates do not always understand her writings, since the Text-to-Pictograph translation tool often generates incomplete and erroneous output when they attempt to translate her messages into pictographs. Chara's mother installed a spell checker on her computer, but Chara finds the spelling suggestions rather confusing. Even more, the spell checker often does not propose a correct alternative at all.

4.1 Status Quæstionis: Spelling Correction of Microtext

The rapid dissemination of electronic communication devices has triggered the emergence of new forms of written texts (Kobus et al. 2008). Microtext, or chatspeak-style text, such as tweets or text messages, is notorious for its many abbreviations, misspellings, phonetic text, colloquial and ungrammatical language, lack of punctuation, and inconsistent capitalisation (De Clercq et al. 2013).

Over the past few years, several linguistic models and algorithms have been proposed to deal with many different types of spelling errors. We will focus on three popular models for the correction of microtext in particular, as proposed by Kobus et al. (2008): the noisy channel model, the machine translation model, and the speech recognition model.

The concept behind the **noisy channel** model, also called the *spell checking* model, is to consider a spelling error as a noisy signal that has been distorted somehow during transmission (Bassil & Alwani 2012). The noisy channel model applies spelling correction on a word-per-word basis and is usually limited to the correction of out-of-vocabulary (OOV) words, relying on orthographic or phonemic surface similarity between two forms. Examples of the noisy channel approach for spelling correction are the rule-based system developed by De Neef & Fessard (2007), the system incorporating phonetic information developed by Toutanova & Moore (2002), and the hidden Markov model developed by Choudhury et al. (2007), which handles both graphemic and phonetic variants. Beaufort et al. (2010) note that the noisy channel model places excessive confidence in word boundaries.

The **machine translation** (MT) model considers the ill-formed text as the source language, and the correct text as the target language. As in general statistical MT, a translation model is trained on parallel data, which is then combined with a language model to transform the noisy input into a string that is closer to the standard (Schulz et al. 2016:4). Aw et al. (2006), for example, use phrase-based MT to tackle the spelling correction problem. It should be noted, though, that constructing an annotated corpus to cover ill-formed words and context-appropriate corrections is labour-intensive (Han & Baldwin 2011), especially since the lexical creativity in microtext is difficult to capture. Another issue is the fact that statistical machine translation allows to handle many-to-many correspondences and applies methods to model the possible mismatch in word order (Kobus et al. 2008), while the normalisation task is almost deterministic (Beaufort et al. 2010), with no change in word order.[6] De Clercq et al. (2013) implement an MT-based approach and describe the first proof-of-concept system for Dutch user-generated content normalisation, but they do not consider users with ID.

The **speech recognition model** converts the input string into a phone lattice and creates a word-based lattice using phoneme-to-grapheme rules. A language model is used to identify the most probable combination of variants (Beaufort et al. 2010). An example of this method is presented by Kobus et al. (2008). Han & Baldwin (2011) identify normalisation candidates for OOV words by decoding the pronunciation of all in-vocabulary words into phonemes and retrieving all words that lie within a threshold character edit distance between the pronunciation of the OOV words and the pronunciation of the dictionary words.

We should remark that, although often presented as such, these three traditionally defined approaches toward microtext correction are not mutually exclusive. For instance, language models are used both in machine translation and speech recognition approaches, while the idea behind noisy channel models underpins statistical machine translation models (by mapping words or phrases onto their translations) and speech recognition models (by mapping words or phonemes onto acoustic waveforms).

The spelling correction system that is presented in this chapter can be considered as a combination of all three approaches, but also introduces new ideas. Although not

[6] Note, however, that Kobus et al.'s argument is invalid when the distortion limit parameter of the machine translation system is set to zero (i.e., monotone translation, without re-ordering).

only OOV words are considered, we generate spelling variants for individual tokens, on a word-per-word basis (noisy channel model). More specifically, the variants are generated (in the first place) by considering the ill-formed word as a result of phonetic confusion (speech recognition model). Finally, we match our new spelling hypotheses against a target language corpus of correctly written text (machine translation model and speech recognition model). Our approach does not require large amounts of annotated data, making it suitable for other languages or normalisation tasks for which very few or no parallel data are available.

4.2 Error Distribution: Comparison with Tweets

Spelling correction for microtext being a fairly young domain of research, its focus lies on users who do not necessarily have a disability, using data from mainstream social media websites. However, many people with ID resort to specialised communication platforms and apps, such as the WAI-NOT environment. The spelling correction tool possibly needs to deal with a completely new and different type of microtext. To verify this, we present a comparison between tweets that are written by people who (supposedly) do not have ID and emails that are sent with the WAI-NOT email client.

Text type	# Non-word errors	# Real-word errors	# Words	% Errors
WAI-NOT	481	183	8077	8.2%
Tweets	182	88	10964	2.5%

Table 4.2: Total amount of misspelled tokens.

We collect a total of 1,000 subsequent tweets from the Dutch Twitter feed (by using the Dutch language filter), having excluded those messages that were not personal, such as news articles or advertisements. Additionally, a total of 1,000 random WAI-NOT emails are selected (see section 2.2.4.2).[7] We manually correct all tweets and email messages and present the error distribution.

Generally speaking (see Table 4.2), many more errors are found in the WAI-NOT messages (8.2%) than in tweets (2.5%). Note that both OOV words and real-word errors are considered.

[7]A total of 1,379 random messages were selected, but 49 messages were completely unreadable and 330 messages were pictograph-based.

Text type	Total # misspelled	# Phonetic words	% Phonetic words
WAI-NOT	664	346	52.1%
Tweets	270	95	35.2%

Table 4.3: Total amount of misspelled words that are a phonetic approximation of the correct word.

As shown in Table 4.3, the majority (52.1%) of spelling mistakes that are made by people with ID is caused by phonetic confusion. Phonetic confusion is the orthographic approximation of a word's pronunciation (such as *wiekent* for *weekend*). Although this phenomenon can also be observed in tweets (35.2%), phonetic spellings of Twitter users tend to show more regularity, at first sight. We were able to spot deliberate mistakes in an attempt to mimic speech (such as the final *t* deletion in *da* or *nie* for *dat* 'that' and *niet* 'not'), and recurrent grammatical mistakes (such as *jou* 'you' versus *jouw* 'your' or *gebeurt* 'happens' versus *gebeurd* 'happened'). Han & Baldwin (2011:368) note that ill-formedness in regular messages is often intentional, whether due to the desire to save characters or keystrokes, due to the wish to belong to a social group, or due to convention. Phonetic mistakes in WAI-NOT messages, on the other hand, are most likely undeliberate mistakes in an attempt to produce correctly written text, and are therefore much more diverse. This idea is reinforced by the fact that a large part of the analysed messages are addressed at teachers or caregivers, for whom the users might make a deliberate effort.

As an additional error measure, we count the number of insertions, deletions, and substitutions needed to get from the original messages to their corrected counterparts (see Table 4.4). On the average, messages in WAI-NOT require 1.4 character operations per erroneously spelled word, while tweets require 1.7 operations. This difference[8] can be explained as follows. Relatively speaking, Twitter users are much more likely to delete characters (more specifically, 75.6% of all required character operations are insertions) than WAI-NOT users (48.6%). This observation is most likely due to the character limit in tweets or a desire to belong to a social group. Examples of deliberate abbreviations in tweets that require many character insertions are *wrschnlk* for *waarschijnlijk* 'probably' (6 character insertions) and *mssch* for *misschien* 'maybe' (3 character insertions).

[8]Note that this difference is not statistically significant according to the *t*-test. This test allows us to compare the averages of two separate populations or groups (in this case, WAI-NOT users and Twitter users).

	Levenshtein distance	# Words	Percentage
WAI-NOT	1	479	72.1%
	2	128	19.3%
	3	44	6.6%
	4	9	1.4%
	5	2	0.3%
	6	2	0.3%
Tweets	1	166	61.5%
	2	66	24.4%
	3	19	7%
	4	7	2.6%
	5	4	1.5%
	6	4	1.5%
	7	2	0.7%
	8	2	0.7%

Table 4.4: Overview of amount of character operations required per erroneously spelled word.

There exist other problems related to spelling errors that may need correction (Han & Baldwin 2011:371) (see Table 4.5). Flooding, the constant repetition of one character, which occurs when emphasis is given by the user (such as *noooooo* or *coooool*), can be found in both genres, but we did not encounter any examples of numbers encoding phonetic values (such as *m8* for *mate* in English, or *macht* 'power' in Dutch), in neither of the two genres. English words (as a result of code switching) are hardly used in the WAI-NOT messages (with the exception of two occurrences of *I love you*), and abbreviations (such as *m.b.t.* 'w.r.t.' for *met betrekking tot* 'with respect to') do not occur at all. Therefore, as long as our spelling correction tool focusses on users with ID, it should not deal with foreign language detection or abbreviation solving.

Text type	**# Flooding**	**# Phonetic numbers**	**# English words**	**# Abbreviations**
WAI-NOT	10	0	6	0
Tweets	9	0	72	59

Table 4.5: Other factors that should be taken into consideration.

From this analysis, it can be concluded that text written by people with ID is, indeed, a different type of microtext. Not only does it contain more errors and do WAI-NOT users rely more on phonetic approximations than Twitter users do, common abbreviations are lacking, and English words are almost never used when a Dutch alternative is available.

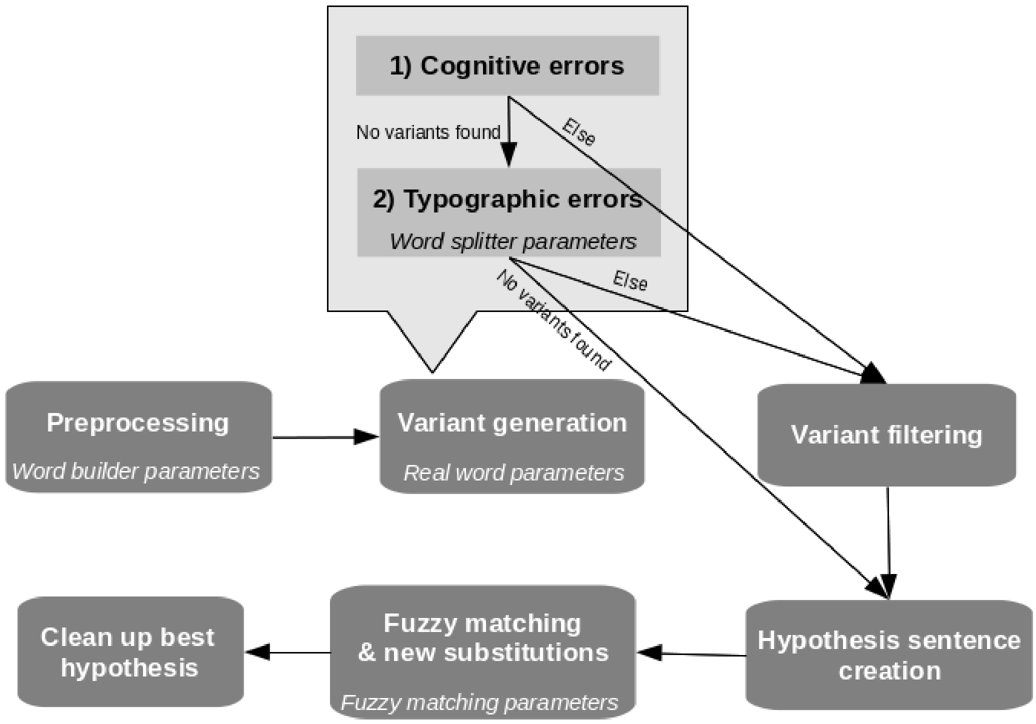


Figure 4.2: System architecture of the spelling correction tool.

4.3 System Architecture

We describe a prototype version of a spelling corrector that is specifically tailored to Dutch text written by people with ID (see Figure 4.2). In the first phase, the input text to be corrected undergoes a number of pre-processing steps (section 4.3.1). Next, spelling variants are generated for OOV tokens and infrequent real words (section 4.3.2); if more than one variant is found, a language model is used to narrow down the total amount of possibilities. We apply character-based fuzzy matching to find the best combination of spelling variants and, additionally, perform new character substitutions when a strong context match is found (section 4.3.3). Finally, we describe the parameter tuning process (section 4.3.4).

4.3.1 Pre-processing

The input text undergoes a number of pre-processing steps.

The *word builder* (Vandeghinste 2002) takes every two adjacent tokens and checks whether they can be merged to form a real word. The word builder parameters (see section 4.3.4) set different threshold frequencies for the (non-)acceptance of the newly created compound word.

The rule-based *tokeniser* splits the punctuation signs from the words, as the variant generation module works on the token level. Given that the hyphen/dash and the apostrophe often belong to the word, they are not considered as separate tokens.

Although most messages that are sent by the users only contain one sentence, *sentence detection* is applied. Segmentation is based on full stops.

In the next step, upper-case letters are converted to lower-case letters. Names retain their capital first letter. They are not involved in the spelling correction process, as long as the name can be found in the database of first names.[9]

The constant repetition of one character or *flooding* is tackled by reducing any repeated sequence of characters to two characters.[10]

Finally, we create a very small dictionary containing popular greetings (such as *hey*) for tokens that have to be left out of the correction process, as these tokens do not always occur in the lexicon.

4.3.2 Variant Generation and Filtering

Spelling variants are generated for all OOV tokens and infrequent real words according to a lexicon.[11] The variant generation process focusses on phonetic, i.e., cognitive

[9] See footnote 3.

[10] While words like *zeeën* ‘seas’ exist in Dutch, *ë* and *e* are considered different characters in our approach, and these tokens will not be subject to the flooding treatment. However, if a user would misspell *zeeën* as *zeeen*, the repeated sequence of characters would be reduced to two characters, due to the flooding treatment. The missing character *ë* is introduced in the variant generation phase, giving rise to the correct variant *zeeën* ‘seas’ - although other, unwanted variants, such as *been* ‘leg’ or *zeep* ‘soap’, could be generated in the variant generation phase, as well. A more elegant solution to this problem would be to check for any sequence of three subsequent vowels (such as *eee*, *iee*, and *aie*) whether a diaeresis could be added to the second or third vowel, and whether a real-word variant could be created this way.

[11] We used the freely available and open lexicon from www.opentaal.org, the Dutch lexicon used in spell checkers such as Hunspell, the open-source spell checker for OpenOffice and MacOS.

errors (section 4.3.2.1). The system also checks for basic typographic errors (section 4.3.2.2). The total amount of variants is narrowed down by a trigram language model before proceeding to the next step (section 4.3.2.3).

4.3.2.1 Generating Variants for Cognitive Errors

Cognitive errors occur when the writer does not know how to spell a word, and relies on the identical pronunciation of words (Toutanova & Moore 2002). As shown by the error distribution in section 4.2, phonetic confusion causes the majority of spelling errors that are made by the target group (see, for example, the misspelled word *wiekent* in Figure 4.1).

4.3.2.1.1 Building the Conversion Rules The approach described in this section is partially inspired by the finite-state framework for normalising SMS messages developed by Beaufort et al. (2010).

First, we manually correct the 1,000 sentences written by the WAI-NOT users (see section 4.2).

We then align the uncorrected and corrected sentences on the character level, using Levenshtein Distance Alignment.[12] This metric (Levenshtein 1966) computes the edit distance of two strings by measuring the minimum number of operations (substitutions, insertions, deletions) required to transform one string into the other. Delimiters, such as commas and spaces, are also aligned. Missing characters on either side of the alignment are indicated by inserting a hyphen (-).

In the next step, we create token pairs. A token pair is retrieved when the same delimiter is found at the same location in the source language/uncorrected and target language/corrected character string. From these token pairs, we extract all possible 4-gram character alignments. We repeat this process for trigrams, bigrams, and unigrams.

Having obtained all 4-gram, trigram, bigram, and unigram character alignments, probabilities are estimated: For every character sequence on the source side, we calculate the likelihood of obtaining an identical or different character sequence on the target side. For example, the character trigram *int* on the source language/uncorrected side was found to be correct and remained *int* in 91% of the cases, but it had been manually corrected into *ind* in the remainder of the sentences. We throw away alter-

[12] http://rosettacode.org/wiki/Levenshtein_distance/Alignment

n-gram	Original sequence	Possible variants
4-gram	djes	djes tjes
	orie	orie orry
Trigram	lls	lles ls
	ort	ord ordt ort
Bigram	gt	cht gd gt
	ni	ni nie niet
Unigram	l	l ll

Table 4.6: Some examples of character sequence rewrite rules.

nations that have a chance of occurrence that is lower than 1%.[13] The idea behind this is that rarely occurring alternations might actually have a typographic, rather than a phonetic origin. By contrast, more commonly occurring alternations are less likely to be random and can probably be attributed to phonetic confusion.

This inventory of commonly appearing alternations allows us to build a system of character rewrite rules (see Table 4.6), in which character 4-gram rules overrule trigram rules, trigram rules overrule bigram rules, and so on.

4.3.2.1.2 Applying the Conversion Rules For every non-word and every real word that has a frequency that is lower than the *real word minimum frequency threshold* (see section 4.3.4) in the frequency list,[14] the conversion rules are applied (see Figure 4.3). A four-character window slides over the token, starting with the first four characters of the token, and verifies whether a 4-gram rule can be found for that character se-

[13]This threshold of 1% was determined by manually checking 300 random cases.
[14]See footnote 4.

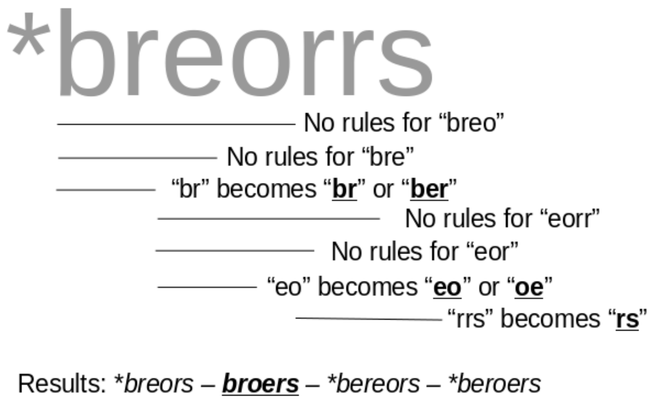


Figure 4.3: Example of how the conversion rules are applied to an OOV token.

quence. If a rule is found, all possible conversion outputs (including the original sequence) are stored and the system proceeds to check the next four characters of the token. If no rule is found, the system backs off to the first three characters of the token and attempts to find a trigram rule for that character sequence. If no rules are found, the first character is retained and the system attempts to find rules for the following four-character sequence.[15]

Finally, all the output sequences are concatenated, and at this point, both non-words and real words may have been formed. If a real word is formed with a frequency that is higher than the *variant frequency* (see section 4.3.4), i.e., the minimum frequency threshold, it is retained as a variant for that token.

4.3.2.2 Generating Variants for Typographic Errors

Typographic errors are mostly related to the keyboard (Toutanova & Moore 2002). If the cognitive error correction module was not able to generate variants for a non-word or a word that has a frequency lower than the *real word minimum frequency threshold* (see section 4.3.4), basic typographic error correction principles are applied. Variants are generated based on five different operations.

[15] Note that this module has not yet been optimised. For instance, in Figure 4.3, the 4-grams *reor* and *orrs* are not checked. One way this system could be improved would be by looking up all the higher-order *n*-gram character sequences in the token before backing off to the lower-order *n*-gram rules. Applying more lower-order *n*-gram character sequence substitution rules, instead of applying less higher-order *n*-gram character sequence substitution rules, unavoidably leads to a greater amount of character sequence variants to be generated, which could cause combinatorial explosion.

The first operation is the *word splitting* module. This is an insertion module for one space character. The system checks whether the erroneous or infrequent token can be split into two parts at any position. Frequency thresholds are determined by parameters (see section 4.3.4).

The next operations are a *one-character deletion*, *insertion*, *substitution*, or *adjacent transposition* at every position of the token. If a real word is formed with a frequency that is higher than the *variant frequency* (see section 4.3.4), it is retained as a variant for that token.

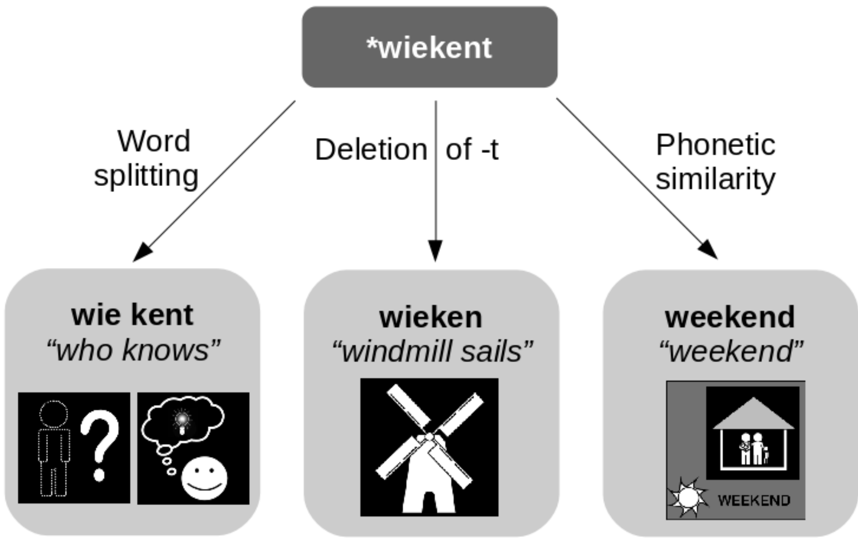


Figure 4.4: A misspelled word like *wiekent* could give rise to at least three variants.

4.3.2.3 Filtering the Variants

At this point, the system may have generated multiple variants for a single erroneous word or infrequent real word (see Figure 4.4). This is usually the case when the input word is short. The filter module attempts to narrow down the total amount of variants before proceeding to the next step: A trigram (word-level) language model trained on a large corpus (a combination of the Dutch part of Europarl (Koehn 2005), Corpus Gesproken Nederlands (Oostdijk et al. 2002), Cross-Language Evaluation Forum (Peters & Braschler 2001), DGT-Translation Memory (Steinberger et al. 2012), and Wikipedia[16]) is used to check whether there is a *possibility* for the variant to occur within its direct context. As the token's direct context may also contain variants,

[16] https://www.wikipedia.org/

all possible trigrams are checked. If a match is found, the variant is retained. If no trigram, bigram, or unigram matches are found for any of the variants (because the context does not provide enough information), all variants are retained and taken to the next step.

The next objective is to identify the most probable variant for each token.

4.3.3 Character-Based Fuzzy Matching

Fuzzy matching techniques allow to find strings in a corpus that approximately (rather than exactly) match a string. We are applying this technique to a monolingual corpus. In the development of the spelling correction tool, we explore the possibility of applying **fuzzy matching techniques** at the **character level** (instead of the word level or the phrase level, as is usually the case in machine translation).

The combination of all variants leads to the creation of a number of potentially correct sentences. Each of these sentences is a hypothesis, one of which will receive the highest score through fuzzy matching. Every hypothesis is treated as a sequence of unigram characters; in other words, in our approach, the unigram is a single character. The space is replaced by a dummy character, the % sign, and is also treated as an individual character. For the monolingual corpus, we use the Corpus Gesproken Nederlands (Corpus of Spoken Dutch, (Oostdijk et al. 2002)), since spoken language better reflects the language used in user-generated content (De Clercq et al. 2013). The sentences in this corpus are also treated as sequences of unigram characters.

In the fuzzy matching process, we use a filter called *approximate query coverage* (Vanallemeersch & Vandeghinste 2015). Its purpose is to select candidate sentences in a corpus which are likely to reach a minimal matching threshold when submitting them to a fuzzy matching metric, in order to increase the speed of matching. Candidate sentences share one or more *n*-grams of a minimal length (see section 4.3.4) with the input hypothesis, and share enough *n*-grams with the input hypothesis to cover the latter sufficiently (according to some threshold, see section 4.3.4). A very efficient search for sentences sharing *n*-grams with the input hypothesis can be done by means of a suffix array (Manber & Myers 1993).[17]

A hypothesis that shares many and long character *n*-grams with candidate sentences from the corpus has a higher likelihood of becoming the winning hypothesis than one that shares only few and short *n*-grams.

[17]We used the SALM toolkit (Zhang & Vogel 2006) for building and consulting suffix arrays.

Our context-sensitive fuzzy matching method also allows us to deal with additional spelling errors, even if a correct variant had not been generated in the variant generation phase. If a high-scoring corpus match is found for two strings of characters, and there is a gap of maximum three characters between those strings in both the corpus and the original hypothesis, we replace these characters in the hypothesis by the ones that are found in the gap in the corpus. For example (see Figure 4.5), the hypothesis *kan je dat misschien nog aan jou moeder vragen* 'maybe you can ask your mother' contains a commonly made spelling mistake. *Jou* is a personal pronoun, whereas *jouw* is a possessive pronoun. In this case, *jouw* would be correct. However, no variants were generated for *jou* in the variant generation process, as it is a highly frequent real word.

One of the matching strings for this hypothesis in the corpus is *nu moeten we het nog aan jouw moeder vragen* 'now we need to ask your mother'. There is a strong overlap between the hypothesis and the corpus match. The system finds a one-character gap containing the character *w* in the monolingual corpus, surrounded by the two matching substrings of minimal length *n* (see section 4.3.4). The character is inserted in the hypothesis, and the spelling mistake is corrected.

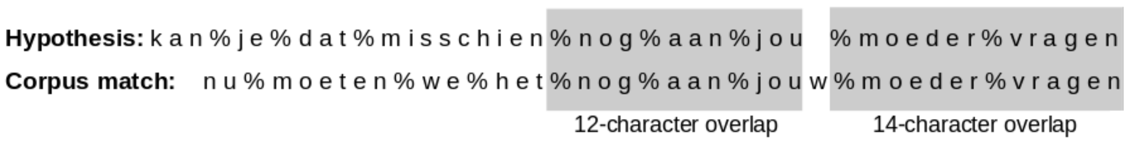


Figure 4.5: Example of a high-scoring corpus match that contains two long, overlapping character strings with a gap of one character.

The winning hypothesis is cleaned up: The spaces between the characters are removed, the % signs are converted into spaces, and the first letter is capitalised.

4.3.4 System Parameters

The system contains a number of tunable parameters.

There are two *word builder parameters*. The first parameter concerns the frequency of the separate parts (for instance, *hand* 'hand' and *schoen* 'shoe') of the (potentially in-vocabulary) compound token (for instance, *handschoen* 'glove'). If both parts are real words and their frequency (according to the lexicon) is high enough, they do not have to pass through the word builder module. The other parameter is the minimum frequency (according to the lexicon) required to accept a newly built real word.

Similarly, there are two *word splitter parameters*. The first parameter concerns the minimum frequency of the token. If the frequency of this token is high enough, it will not have to pass through the word splitter module. However, if the frequency is not high enough or if the token turns out to be a non-word (for instance, *naarhuis* in Figure 4.1), the system will attempt to split the token into two real word parts (for instance, *naar* 'to' and *huis* 'home'). The second parameter sets a minimum frequency for those two words. If the frequency is high enough, the original word is split.

Parameter	Min	Max	Step	Value
Word builder parameters				
Separate word min. frequency	1	1000	10	120
Compound word min. frequency	1	3000	10	210
Word splitter parameters				
Separate word min. frequency	1	1000	10	1,680
Compound word min. frequency	1	3000	10	1,760
Real word parameters				
Real word min. frequency	1	3000	10	100
Variant min. frequency	1	500	10	220
Fuzzy matching parameters				
n-gram parameter	6	12	1	8
Minimum score parameter	0.2	1	0.1	0.2
Highest frequency threshold	1	500	10	100

Table 4.7: Parameter values for the spelling correction module after tuning.

The *real word minimum frequency threshold* determines how common a correctly spelled word should be in order to avoid going through the spelling variant generation process. When real word variants are generated for a token, they need to have a minimum frequency, the *variant frequency*, in order to be accepted as a variant.

There are also three fuzzy matching parameters. The *n-gram parameter* decides on the minimum amount of contiguous characters that should occur as a sequence in the corpus sentence in order for the fuzzy match to be successful. The *minimum score parameter* sets the minimum matching score needed to retrieve a sentence from the monolingual corpus. The *highest frequency threshold* decides that, if a certain *n*-gram has a very high frequency, the fuzzy matching system will ignore it for fuzzy matching.

We create a tuning corpus by manually correcting 200 WAI-NOT messages that were not used during the development of the system. We tune the parameters by means of an automated procedure, using a local hill climber (Vandeghinste et al. 2017), which varies the parameter values when running the spelling corrector script on the tuning corpus. The BLEU metric (Papineni et al. 2002) is used as an indicator of relative improvement. We run five trials of a local hill climbing algorithm, until BLEU converges onto a fixed score. Each trial is run with random initialisation values, and we vary the values between certain boundaries in order to cover different areas of the search space. From these trials, we take the best scoring parameter values. These values are presented in Table 4.7.

4.4 Evaluation

In this section, we present the results of our evaluations. Section 4.4.1 evaluates the system on a test set of WAI-NOT messages and compares this output to the output of the baseline spelling correction system. Section 4.4.2 evaluates the system within the larger context of the Text-to-Pictograph translation pipeline.

4.4.1 Intrinsic Evaluation

After having filtered out noise and pictograph-based messages, we randomly select 300 emails from the WAI-NOT corpus that were not used during the development of the system, and we manually correct them to the best of our ability. The lower bound is the original set of uncorrected messages. Table 4.8 shows the word-based **BLEU**, **NIST** (Doddington 2002), and **Word Error Rate (WER)** scores.[18] We also calculate the amount of **character operations** needed in order to get to the reference correction. Significance levels are calculated for BLEU and NIST by comparing each system with the system on the previous line using bootstrap resampling (Koehn 2004).

As shown in Table 4.8, the baseline spelling corrector does more things wrong than right.[19] However, significant improvements are made using the new corrector.

[18]We do not consider the Position-independent word Error Rate (PER), because there is no change in word order.

[19]These results contradict the results in Table 3.2, where the automated spelling corrector conditions, which uses the old corrector, perform significantly better than the conditions in which no spelling corrector is used (in the more advanced conditions, such as *synonyms* and *relations*). We dug deeper and evaluated the effect of the spelling corrector on the WAI-NOT test set without translating them

System used	BLEU↑	NIST↑	WER↓	# Character operations
No corrector	64.02	8.62	12.37	699
Old corrector	62.88**	8.07**	19.51	816
New corrector (language model)	78.72**	10.23**	8.02	298
New corrector (fuzzy match)	**84.04****	**10.49****	7.57	238

Table 4.8: Automated evaluations on 300 email messages. $*p < 0.05, **p < 0.01$.

To verify the effects of the character-based fuzzy matching method, as opposed to more conventional methods toward automated spelling correction, we replace the fuzzy matching step with beam search decoding on a 5-gram language model with interpolated modified Kneser-Ney smoothing (Chen & Goodman 1999), trained with the KenLM Language Model Toolkit (Heafield et al. 2013) on a very large mixed-domain corpus.[20] The difference in performance between the two systems is statistically highly significant, with the fuzzy matching-based corrector outperforming the beam search decoder.

Tables 4.9 and 4.10 present a more fine-grained analysis of how the baseline system and the new system deal with erroneous words.

With respect to **recall** (TP divided by TP+FN), the new system is able to propose a correction for 64.21% of the erroneously spelled words, whereas the baseline system is able to retrieve a variant for 42.28% of the erroneously spelled words. Regarding **precision** (TP divided by TP+FP), 100% of all words that are considered to be erroneous by the new corrector are, indeed, erroneously spelled words. Precision in the old system is 87.79%; it corrects some words that should not have been corrected in the first place. Erroneous corrections concern popular greetings like *hey*

into pictographs. The lower bound, the original set of uncorrected messages, reaches a BLEU score of 63.49. After applying the old spelling corrector, BLEU reaches a score of 65.51. This result can be attributed to the characteristics of the test set. We manually checked the corrected WAI-NOT test set and found that there are no false positives which could affect the BLEU score negatively. No inappropriate changes were made to proper names, as they all appear in our list of first names, or simply because no real-word alternatives were found. Furthermore, the WAI-NOT test set was constructed manually by Vandeghinste & Schuurman (2014), while the 300 sentences that we used for the evaluations in Table 4.8 were generated automatically. With a primary objective of measuring pictograph translation quality, the WAI-NOT test set does not contain sentences that are nearly unreadable. This observation explains why the old corrector performs slightly better on that test set.

[20] A more detailed description of this method and the mixed-domain corpus is given within the context of Pictograph-to-Text translation (see section 9.2.2).

and proper names that are not included in our list of first names and for which the system attempts to generate a low-frequency variant. These problems are solved in the new system by the introduction of a small greetings dictionary and the fact that low-frequency variants are not proposed.

Of all the erroneous words that are detected by the new system, 83.3% (145 out of 174 cases) is replaced by its correct variant, while the baseline system successfully corrects 35.6% (41 out of 115 cases) of all detected words. The new system is able to detect more erroneous forms (i.e., there are less false negatives), as the baseline system was limited to OOV errors. Note that 71.4% of the unretrieved words in the new system are highly frequent real words. Closer inspection reveals that the large majority of these real-word errors can be attributed to grammatical mistakes, such as the confusion between the personal pronoun *jou* and the possessive pronoun *jouw*.

	Treated as positive (i.e., erroneous; corrections are made)	**Treated as negative** (i.e., correct; no corrections are made)
Really is positive (i.e., erroneous)	**TP** 115 *(41 correct + 74 incorrect corrections)*	**FN** 157 *(81 non-words + 76 real words)*
Really is negative (i.e., correct)	**FP** 16	**TN** 2670

Table 4.9: Confusion matrix (old corrector).

	Treated as positive (i.e., erroneous; corrections are made)	**Treated as negative** (i.e., correct; no corrections are made)
Really is positive (i.e., erroneous)	**TP** 174 *(145 correct + 29 incorrect corrections)*	**FN** 97 *(28 non-words + 70 real words)*
Really is negative (i.e., correct)	**FP** 0	**TN** 2700

Table 4.10: Confusion matrix (new corrector).

Comparing our system with other systems is difficult, as they do not target text written by users with ID (and most tools focus on English text). De Clercq et al.

(2013), who created the first normalisation tool for Dutch microtext, admit that words requiring different types of operations are difficult for their system, whereas our approach allows for multiple substitutions within a single word. De Clercq et al. show that their system is best at resolving smaller words requiring only one or two insertions, while phonetic problems, in particular, turn out to be an obstacle for their tool.

In a follow-up project to De Clercq et al. (2013), Schulz et al. (2016) present a multimodular approach to account for the diversity of normalisation issues encountered in Dutch user-generated content. Their system was developed around the same time as our correction tool. Just like our system, its architecture consists of three layers: a pre-processing layer (sentence splitting, tokenisation, and correction of flooding), a suggestion layer, and a decision layer, which chooses the best combination of variants from the pool of suggestions. The suggestion layer comprises a variety of modules that have been conceived to account for the different normalisation issues encountered, including a word splitting and compounding module, a Levenshtein-based spell checker, and a transliterate module,[21] which is in many ways similar to our cognitive variant generation module. The decision layer in Schulz et al. (2016)'s system is different from ours, as it makes use of a language model, instead of character-based fuzzy matching techniques. The oracle values in Schulz et al. (2016)'s study show that the decision module obtains a high performance despite the large number of suggestions generated by the suggestion layer, with precision values of up to 93.4% (as compared to a baseline of 86.6%, i.e., uncorrected input) and recall values of up to 92.9% (as compared to a baseline of 84.1%) on a mixed corpus of short messages (SMS), tweets, and social media posts.

4.4.2 Extrinsic Evaluation

We manually evaluated the effects of the spelling correction system within the larger context of the Text-to-Pictograph translation tool. The baseline, which uses the old spelling corrector, is described and evaluated in section 3.3. The new system implements the improved spelling corrector.

The evaluation set consists of 50 Dutch messages that have been sent with the WAI-NOT email system (84 sentences or 980 words) (see Appendix B.2). We trans-

[21] A supervised machine learning classifier in which each grapheme in the non-normalised input sequence is associated with a class in the output sequence.

		Without proper names		With proper names	
Condition	**Precision**	**Recall**	**F-score**	**Recall**	**F-score**
Sclera					
Baseline	89.2%	86.2%	87.7%	85.2%	87.2%
New system	92.6%	89.1%	90.8%	88.2%	90.3%
Rel.improv.	*3.7%*	*3.3%*	*3.5%*	*3.6%*	*3.6%*
Beta					
Baseline	85.9%	89.5%	87.6%	88.7%	87.3%
New system	89.8%	91.5%	90.6%	90.8%	90.3%
Rel. improv.	*4.5%*	*2.3%*	*3.4%*	*2.4%*	*3.5%*

Table 4.11: Manual evaluation of the Text2Picto translation engine. The baseline uses the baseline spelling corrector, whereas the new system uses the improved spelling corrector.

lated these messages into a sequence of Sclera or Beta pictographs using the Text-to-Pictograph translation tool.

We perform a manual annotation, removing untranslated words that are considered not to contribute to the content. This allows calculating the **recall**. For each of the translated words, we judge whether the pictograph generated is the correct pictograph, in order to calculate **precision**. Results are presented in Table 4.11. We calculate recall and **F-score** with and without proper names, in the latter case removing all proper names from the output. Precision remains the same in both conditions.

Figure 4.6: Example of correct Text-to-Sclera translation.

An increase in precision and in recall is obtained for both the Beta and the Sclera condition. Examples of erroneous words that previously could not be translated into pictographs are *grapeg* for *grappig* ‘funny’, *ikhoop* for *ik hoop* ‘I hope’ and *heeeeel* for *heel* 'very'. Examples of erroneous words that previously led to an erroneous pictograph translation were *wiekent* for *weekend* 'weekend', which was erroneously corrected into *wieken* 'wings' (and translated into a pictograph showing a bird's wings, see also Figure 4.1 and Figure 4.6), and *moelijke* for *moeilijke* 'difficult', which was erroneously corrected into *mogelijke* 'possible' (and translated into a pictograph showing the verb "may/can").

4.5 Conclusion: Automated Spelling Correction for People with ID

We described an automated spelling corrector for Dutch text written by people with ID, the first of its kind. The system can be extended to other languages and our methods can be re-used for other tasks,[22] provided that some corrected data are available in order to infer phonetic rules for the variant generation step. Nevertheless, our approach requires only a small amount of training data. The results show that the new system significantly improves over the baseline, but there is still some room for improvement.

In the first place, the variant generation process is not yet able to correct tokens which present elements of phonetic confusion and typographic errors at the same time. For the time being, the cognitive error variant generation step and the typographic error variant generation step are two unrelated steps within the variant

[22]We verified the extensibility of our approach in a shared task aimed at the normalisation of 17th-century Dutch text (Tjong Kim Sang et al. 2017). Historical texts pose a challenge for automatic text processing tools, such as part-of-speech taggers, because the words in the text are spelled in a different way in comparison with their modern equivalents, and because spelling may be inconsistent. Our approach for translating 17th century to modern Dutch is fully data-driven and makes use of parallel text (the parallel 1637 and 1888 Bible versions, which were provided for the shared task) and a monolingual corpus of modern Dutch (a 1-million sentence mixed-domain corpus, extracted from Opus (Tiedemann 2009)). We evaluated our approach using 500 test sentences extracted from "Het journaal van Bontekoe". Our evaluations showed that the spelling rules from our spelling correction module, in particular, were very helpful, and that performing machine translation as an initial step did not lead to improvement, possibly because there were not enough parallel data available. Measured by part-of-speech accuracy of a standard tagger applied to small snippets of text (1,100 to 1,400 tokens) from different authors and from different time periods, in comparison with a gold standard of 87.50%, our system reached an accuracy of 82.50%, whereas the baseline of tagged unmodified text reached an accuracy of 70.90%.

generation process. The ideal scenario would be to find an elegant way to combine both modules without overgenerating. A similar observation is made by Schulz et al. (2016), who developed a normalisation tool for Dutch user-generated content: Their system struggles with normalising words that contain multiple normalisation problems.

For the fuzzy matching step, we could add more and/or different resources to the monolingual corpus and evaluate their added value on the system's performance. These corpora should be exempt from spelling errors and share as many characteristics with informal text or oral conversations as possible. The corpora should also contain plenty of first-person and second-person forms.

Finally, we should consider performing a grammar check during the spelling correction process in order to detect real-word errors that are left out of the variant generation process because of their high frequency, such as *jou* and *jouw*.

Persona: Chara (After)

Chara is no longer hesitant to post status updates and comments on Facebook, knowing that the automated spelling correction tool will take care of most of her spelling mistakes, making her feel less exposed. Her family and fellow fan community members are able to read all about her daily life adventures on their news feed, while her classmates can use the Text-to-Pictograph translation tool to convert Chara's messages into Beta pictographs. The pictograph translations are more accurate than before, and less words are left untranslated.

CHAPTER **5**

Improvement #2: Syntactic Simplification for Pictograph Translation

The baseline Text-to-Pictograph translation system presents the reader with an almost *verbatim* pictograph-per-content word translation and does not perform deep linguistic analysis. Pictograph translations may be too long or complex, leaving people with ID confused and distracted.[1] In section 5.1, we present some examples of syntactic phenomena that may hamper a target user's understanding of pictograph output. Section 5.2 gives an overview of related work on the topic of syntactic simplification methods for natural languages. Note that, while syntactic simplification for pictograph translation has not yet been studied in the literature, there are some strong similarities with simplification for less resourced languages. We present three different approaches toward automated simplification and compression and evaluate how pictograph translation could benefit from each of them. Section 5.3 presents a number of guidelines and sources of inspiration for creating a syntactically simplified visual language: the Chinese language, the Klare Taal checklist for easy-to-read text, and the Dutch WAI-NOT news messages. These resources allow us to identify "complex" syntactic phenomena to be treated by the pictograph simplification module, which we enumerate in section 5.4. In section 5.5, we present the syntactic simplification module. The resulting simplification system is augmented with an advanced verb group

[1]Previous versions of sections 5.1 to 5.7 appeared in Sevens et al. (2017b) and Sevens et al. (In press).

simplification module, which we describe in section 5.6. All simplification modules can be activated optionally, if deemed necessary. Finally, section 5.7 concludes. The examples that are shown throughout this section originate from the WAI-NOT corpus (see section 2.1).[2]

5.1 Problem Definition

The user tests (see section 3.3.3) revealed a number of issues that may arise when a person with ID attempts to read an automatically generated pictograph sequence. We present a few examples here.

Long input sentences correspond to long output sequences, and the more pictographs are displayed on the screen, the more difficult it is to understand the combined semantics of the pictograph sequence. In the sequence shown below (see Figure 5.1), for example, there are no less than five pictographs (*next*, *year*, *in*, *September*, and *want*) to be interpreted until the reader finally encounters the agent *we*.

Volgend jaar in september willen we samen basket doen in de sporthal van de school.
Next year in September want we together basket do in the sports-hall of the school.
'Next year in September we want to play basket together in the sports hall of the school.'

Figure 5.1: Pictograph sequences that can be difficult to read: long sequences.

Coordinate sentences and **complex** sentences are composed of two or more (main and dependent) clauses. Due to the lack of punctuation signs and grammatical properties in the pictograph output, however, it can be challenging for the reader to determine where one clause stops and where another one begins. For example, in the

[2]The examples were manually corrected and anonymised.

pictograph sequence shown below (see Figure 5.2), it could be unclear to the reader who has been *sending the messages*, and who or what is supposed to be *okay*.

De berichten die je naar mij hebt gestuurd zijn in orde.
The messages that you to me have sent are in order.
'The messages that you have sent to me are okay.'

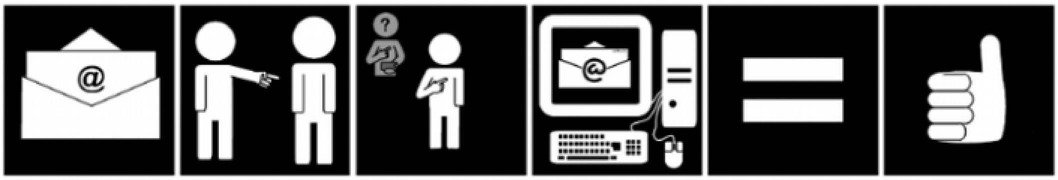


Figure 5.2: Pictograph sequences that can be difficult to read: translations of coordinate or complex sentences.

In Dutch, **subject-verb inversion** occurs in interrogative sentences, imperative sentences, topicalised sentences, and subordinate clauses. It is, therefore, not always safe to assume that the pictograph(s) of the subject will appear directly in front of the pictograph(s) of the verb(s). In the example shown below (see Figure 5.3), the complex pictograph *to give a hug* and the simplex pictograph *you* separate the subject *he* from the finite verb, the auxiliary *to want*, as a consequence of literal translation from Dutch.

Hij zei dat hij u een knuffel wou geven.
He said that he you a hug wanted give.
'He says that he wanted to give you a hug.'

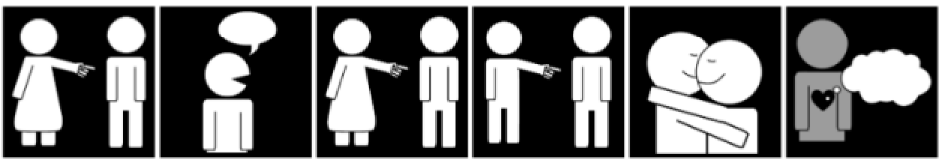


Figure 5.3: Pictograph sequences that can be difficult to read: translations of non-subject-verb-object order.

Another example are **passive sentences**. In passive sentences, the agent of the verb can either be hidden, or it can be overtly expressed by means of a prepositional phrase. Not being able to identify the (correct) agent, due to the lack of prepositions and grammatical properties in the pictograph output, could drastically change the meaning. For instance, the pictograph translation of the passive input sequence shown below (see Figure 5.4) could likewise be interpreted as *I am bullying a girl*.

Ik werd gepest door een meisje.
I was bullied by a girl
'I was bullied by a girl'

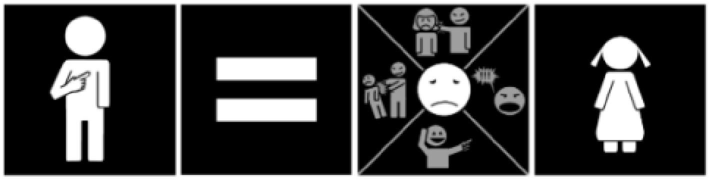


Figure 5.4: Pictograph sequences that can be difficult to read: translations of passive sentences.

We will tackle these, and other syntactic phenomena that are difficult to read, by means of a simplification module that automatically simplifies the input text before Text-to-Pictograph translation takes place.

Persona: Clemont (Before)

Bonnie is studying at a foreign university. While she is enjoying her time abroad, she misses the daily contact with her brother Clemont, who has an autism spectrum disorder and an intellectual disability. Due to speech disorders, telephone calls are difficult for him. Clemont has access to a computer at his day centre, but he is not yet able to read Bonnie's messages without the help of his caregiver. Bonnie is looking for an accessible way to share her stories with Clemont - preferably, without having the need to involve a third party. A few weeks ago, she discovered the Text-to-Pictograph translation technology on the WAI-NOT website, but after translating one of her emails, her screen was flooded with pictographs. She did not consider the translation technology helpful for her brother just yet - except for very short messages, maybe -, and decided to look for better solutions, instead.

5.2 Status Quæstionis: Text Simplification

Text simplification is the process of reducing the linguistic complexity of a text, while still retaining (most of) its original information content (Siddharthan 2014:259). Simplification systems address two different tasks: lexical simplification and syntactic simplification (Medina Maestro et al. 2016:3). *Lexical simplification* is concerned with the substitution of difficult or uncommon words or expressions by simpler synonyms. *Syntactic simplification* is concerned with the simplification of long and complicated sentences into equivalent simpler ones. Given that the Text-to-Pictograph translation system already uses pictographs as a way to replace or support words, we only focus on syntactic simplification in this chapter.

Just & Carpenter (1992:123) argue that many of the mental processes underlying human comprehension occur in parallel. While the comprehender develops the expectation of encountering a verb, he/she also calculates the syntactic, semantic, and pragmatic features of the sentence. All of these processes can be executed simultaneously. However, the more the working memory is required for storing information during a syntactic parse, the less the working memory is available for processing meaning, and vice versa.[3] Complex syntax hinders the processing of meaning.

Long sentences, conjoined sentences, embedded clauses, passive constructions, and non-canonical word order, among other phenomena, increase text complexity for readers with limited literacy skills (Candido Jr. et al. 2009:34). But even for people without a disability, the working memory can limit reading comprehension. Less skilled readers may benefit, for instance, from breaking up complex sentences into several, shorter sentences. This effect can be attributed to the reduction in the amount of information stored in the working memory during syntactic processing, thus freeing up more working memory for higher level semantic processing (Mason & Kendall 1979:11).

Various approaches toward automatic text simplification have been proposed over the years. As Siddharthan (2014:269) notes, some of the insights from the early systems have been rediscovered in recent years, while others have been forgotten. We present some of the most noteworthy systems. The different approaches described below can be largely broken down into two categories: rule-based syntactic simplification (section 5.2.1) and monolingual machine translation approaches (section 5.2.2).

[3] We assume that the working memory executes similar processes during the interpretation of pictograph sequences, although this hypothesis has not yet been tested.

We also consider sentence compression, which is a form of syntactic simplification, although the inverse relation does not hold (section 5.2.3). We then proceed to discuss the merits and disadvantages of each approach for the pictograph translation system (section 5.2.4).

5.2.1 Rule-Based Syntactic Simplification Systems

Chandrasekar et al. (1996) were the first to explore methods to automatically transform long and complicated sentences into simpler sentences, with the objective of reducing sentence length as a pre-processing step for natural language processing applications, such as parsers. Note that, at the time, full parsing was slow and prone to failure, especially on complex sentences. Therefore, syntactic parsers could not be used for simplification. In their first approach, Chandrasekar et al. (1996) view syntactic simplification as a two-stage process: The first stage provides a structural representation for the sentence, on which the second stage applies a sequence of rules. Input sentences are considered to be composed of chunks, or sequences of word groups, such as noun phrases with some attached modifiers. The chunked sentences are simplified using a set of simplification rules, which are manually ordered to take care of more frequent transformations first. However, these rules are not able to deal appropriately with the richness of natural languages. The system often made disambiguation decisions without using a parser in order to be of use to parsing. Therefore, it raised more issues than it addressed (Siddharthan 2006:27). A full syntactic parse, beyond the level of chunking, would be needed to solve these problems. However, using a syntactic parser would defeat the authors' objective, in the first place.

The system described by Chandrasekar & Srinivas (1997) automatically acquires text simplification operations, providing the basis of much contemporary work in the field (Siddharthan 2014:270). In this approach, simplification rules are induced from an aligned corpus of sentences and their hand-simplified forms. The original and simplified sentences are parsed using a lightweight dependency analyser. However, this work did not progress further, and this approach requires the manual simplification of a reasonable quantity of text.

Siddharthan (2006) continues the trend of syntactic simplification without parsers, arguing that long sentences that could benefit from parsers would often time out or result in an erroneous parse. The author proposes a three-stage theory, which decomposes the text simplification task into an analysis stage, a transformation stage, and a regeneration stage. In the analysis stage, a number of shallow linguistic analysis operations are applied. The transformation module is where the actual syntactic simplification takes place, by using a set of simplification rules. Siddharthan presents three rules for conjunction and two rules each for relative clauses and appositive clauses. These rules are incomplete. For instance, there is no mention of relative clauses with a non-subject antecedent, or conversion from passive sentences into active ones. Siddharthan's work includes a very detailed study of text cohesion in the regeneration stage. Using robust and shallow text analysis techniques, and computational models of discourse structure, the author aims to minimise the disruption in discourse structure that could be caused by syntactic simplification. Examples of regeneration tasks are sentence ordering, referring expression generation, and determiner choice.

The Practical Simplification of English Text (PSET) project (Devlin & Tait 1998; Carroll et al. 1999) is the first simplification project aimed specifically at people with aphasia and could, therefore, justify the use of a syntactic parser for the analysis stage, as opposed to the previously discussed systems (Siddharthan 2006:27). PSET uses a probabilistic bottom-up parser for analysing complex input sentences, and a unification-based pattern matching of handcrafted rules for the transformation stage. However, the system only manages to simplify two constructs, namely coordinate clauses and passive voice.

The PorSimples simplification system for Brazilian Portuguese-speaking people with low literacy skills designed by Candido Jr. et al. (2009) analyses a sentence by means of a syntactic parser, which provides a syntactic tree for that sentence, and applies the following six simplification operations in order: clause identification, replacement of discourse markers, conversion of passive voice into active voice, clause re-ordering, conversion to subject-verb-object ordering, and changes in topicalisation for adverbial phrases. The bottleneck of this system, however, is the time needed to parse sentences. Whenever a syntactic operation is applied, the simplified sentence presents a new syntactic structure and must be re-parsed. This is a computationally expensive operation.

Bott et al. (2012) opt for a hybrid approach which largely relies on rule-based components, but integrate data-driven methods where possible. The authors use a corpus of newspaper articles, which are aligned with their manually simplified counterparts, to identify the different types of editing operations that were carried out by human simplifiers. During the pre-processing phase, the authors use a dependency parser. The actual structural simplification is done in two steps. First, a grammar looks for suitable target structures which can be simplified. In the second step, the manipulations are carried out. This simplification system covers the identification of relative clauses, coordinate clauses, and participial constructions.

Finally, Orasan et al. (2018) present FIRST, a tool aiming at making documents more accessible to people with autism spectrum disorder. They develop a method which can automatically identify and classify signs of syntactic complexity using a machine learning approach, and rewrite complex sentences using a predefined set of rules.

Siddharthan (2006:281) notes that handcrafted systems are limited in scope to lexical simplification, but adds that syntactic rules can indeed successfully be written by hand.

5.2.2 Monolingual Machine Translation Approaches to Syntactic Simplification

Siddharthan (2014:279) describes the monolingual machine translation approach to syntactic simplification as a two-stage process. The first step concerns the creation of a phrase table that contains aligned sequences of words in the source and target languages (in this case, original sentences on the source language side, and simplified sentences on the target language side), along with probabilities that indicate the likelihood of phrase translations. The second step is decoding, where the phrase table and a language model of the target language are used to find an optimal translation. Siddharthan notes that phrase-based machine translation can only perform a small set of simplification operations, such as lexical substitutions, deletions, and simple paraphrases. However, they are not well suited for re-ordering or splitting operations. One exception to this is the syntax-based statistical machine translation approach by Zhu et al. (2010), in which relative clauses and appositive clauses are treated by means of a segmentation table that stores the probabilities of sentence splitting at particular words.

5.2.3 Sentence Compression Approaches

The field of sentence compression is a research area that aims to shorten sentences, thus focussing on deletion operations, for the purpose of summarising the main content. Note that, while sentence compression is a form of syntactic simplification, the inverse relation does not hold. Compression filters out less informative portions of a text, while simplification aims to help people achieve a better comprehension of the text.

Two notable examples of sentence compression for Dutch are the systems described by Daelemans et al. (2004) and Vandeghinste & Pan (2004). Interestingly, to our knowledge, these compression tools are also the only systems that have ever been developed for the simplification (in this case, compression, more specifically) of Dutch text.

The system developed by Daelemans et al. (2004), built within the framework of the ATraNoS project, automatically produces subtitles for television programmes on the basis of written transcripts. In their first approach, Daelemans et al. make use of a corpus of full-length sentences aligned with their manually compressed versions (Vandeghinste & Tjong Kim Sang 2004). To this corpus, the authors apply a memory-based learner with 30 features, such as chunk information and lemma information. However, this approach did not perform well and the machine learner frequently made nonsensical errors, most likely because a much larger amount of training data was needed. In their second approach, Daelemans et al. manually compile phrase deletion rules. For example, one such rule states that only the head word of noun phrases is retained. This approach was able to outperform the machine learner.

In a different experiment, Vandeghinste & Pan (2004) use the aligned corpus of Dutch sentences with their manually compressed versions to estimate compression probabilities, such as the removal, non-removal, and reduction probabilities for noun phrases, prepositional phrases, and subordinate clauses. This approach is similar to statistical machine translation using synchronous tree-substitution grammars (Joshi & Schabes 1997). The sentences in the parallel corpus are tagged, chunked, and divided into clauses. However, the analysis step introduced too many errors in the system. Coordinating conjunctions, for instance, were often wrongly analysed, leading to misestimations of the compression probabilities. The authors conclude that a full parse is needed to solve this problem, which was not yet feasible at the time.

Other examples worthy of mention within the field of sentence compression are the work by Woodsend & Lapata (2011), who use quasi-synchronous grammars to generate rewrite operations for a source tree, and the work of Angrosh et al. (2014), who describe a synchronous dependency grammar for text simplification. They use a manually constructed grammar for syntactic rules and an automatically acquired grammar for lexical rules and paraphrases.

5.2.4 Implications for our Approach

While many of the previously described methods have their merits, most of them are not appropriate for the objective of translating Dutch sentences into simplified pictograph sequences. Data-driven machine translation methods require parallel data. The corpus that is needed for our simplification task would have to consist of Dutch sentences (or their pictograph equivalents) on the source language side, and simplified Dutch sentences (or their pictograph equivalents) on the target language side. Not only does such a corpus, to our knowledge, not exist,[4] this approach would not be effective if we want to learn strong simplifications like those performed with the specific needs of our target population. Problems like splitting, for instance, require very complex copying operations.

Rule-based simplification systems do not always require parallel data, which makes them much more suitable for our task. However, when no syntactic parsers are used, rule-based systems are not always able to deal with the richness of natural languages in an appropriate way. Years ago, parsers were not very reliable and would often time out. Today, this argument does not apply anymore. For this reason, we opt for a handcrafted simplification system that makes use of syntactic parsing for sentence analysis.

The compression approach differs from syntactic simplification in that it removes less essential portions of the text, while syntactic simplification still aims to preserve the full information content of the original message. Nevertheless, we expect compression to be a useful tool for readers who are still familiarising themselves with the pictograph set at hand. Therefore, we also describe the development of a simple, optional compression module. Again, as no parallel corpora are available for Dutch, we adopt a rule-based approach.

5.3 Guidelines and Sources of Inspiration for Syntactic Simplification for Pictograph Translation

The following questions arise: What do we define as "complex" syntactic phenomena? And how do we simplify these phenomena in an appropriate way? To answer this,

[4]With the exception of a corpus for sentence compression for Dutch subtitles (Vandeghinste & Tjong Kim Sang 2004). However, as we already remarked, compression is not the same as simplification, and this corpus was developed for deaf people, not for people with ID.

we make use of three sources of inspiration. First, we present the Chinese writing system (section 5.3.1), which shares many of its characteristics with the pictographs that are used in our translation technology: Just like Sclera and Beta, the Chinese logographic system has very few grammatical properties. This leads to a number of decisions on the syntactic level. Next, we discuss the recommendations of Klare Taal (Clear Language) (section 5.3.2). Klare Taal offers a set of guidelines on communicating with people who are (functionally) illiterate by means of clear, coherent, and consistent language. We will use these guidelines as a starting point for the development of a syntactic simplification module. Finally, we analyse a small corpus of handwritten easy-to-read news messages that were published on the WAI-NOT website (section 5.3.3).

5.3.1 The Chinese Writing System

The Chinese writing system is one of the oldest writing systems in the world that is still in use today, with recognisable forms of Chinese characters dating from over 3,500 years ago. Chinese has a logographic system consisting of logograms or characters, which may represent a word, or part of a word. There are four types of characters:

- In some cases, there is an immediate, pictorial connection between the Chinese character and its referent through visual similarity (just like iconic pictographs, see section 1.1.3.1). These characters are commonly referred to as **pictograms**. Examples of Chinese pictograms are *eye*, *tree*, and *woman*. Nowadays, these characters are stylised versions of their ancient forms, although their origins are still recognisable. (see Figure 5.5).

目 木 女

Figure 5.5: Chinese pictograms: *eye*, *tree*, and *woman*.

- The second type is the **ideogram**. These characters are used to represent abstract concepts, such as *up* or *down* (just like symbolic pictographs, see section 1.1.3.3). Also belonging to this category are pictograms with an ideographic indicator. An example of this is the *tree* character with an additional ideographic indicator, which results in the concept *root* (see Figure 5.6).

上　下　本

Figure 5.6: Chinese ideograms and ideographic indicators: *up*, *down*, and *root*.

- **Compound** characters consist of a combination of different elements (just like complex Sclera pictographs, see section 1.3.2.3.1). An example of this is *forest*, which consists of multiple *tree* pictograms (see Figure 5.7).

森

Figure 5.7: Chinese compound pictograms: *forest*.

- **Pictophonetical** characters constitute most of the Chinese logograms. Their meaning component indicates the general meaning of the character, and their sound radical hints at the pronunciation. For instance, different types of trees are depicted by means of a *tree* pictogram and an additional phonetic morpheme (see Figure 5.8).

松

Figure 5.8: Chinese pictophonetical characters: *pine tree*.

Just like Sclera and Beta, the Chinese language almost entirely lacks inflection. Words have only one grammatical form: Number, determiners, gender, and verb tense are typically not expressed by any grammatical means. How, then, does the Chinese writing system compensate for the lack of function words and grammatical markers, and what do these observations imply for the creation of a syntactically simplified pictograph language? We enlist a number of relevant characteristics:

- Basic word order in Chinese is subject-verb-object (SVO). Modifiers precede the words they modify. Fixed word order makes up for the loss of function words or grammatical markers.[5]

[5] Observe that this is a general principle. Languages with a rich morphology tend to have a freer word order than languages that lack such markings, and vice versa.

- While there are no articles, Chinese nouns may be modified by demonstratives, possessives, quantifiers, or numerals.
- Chinese nouns remain invariant in number. Pronouns have singular and plural forms, and gender distinctions for the third person, just like pronouns in Beta and Sclera.
- Chinese uses serial verb constructions, which involve two or more verbs or verb phrases in sequence. Furthermore, auxiliary verbs and modal verbs precede main verbs.
- While tense in Chinese is typically not expressed by any grammatical means, the time at which an action is taking place can be indicated by expressions of time (such as *yesterday* or *tomorrow*), or should be inferred from the context.

Chinese proves that a successful writing system does not necessarily depend on the presence of grammatical markers; it makes up for this loss by using fixed syntactic positions, such as SVO order and the use of serial verb constructions. Invariance of pictograph order within the verb group and within the whole sentence deserves special attention in our approach.

5.3.2 Klare Taal

Companies and organisations could improve their written and oral communication toward non-native speakers and people with limited language skills. Huis van het Nederlands (House of the Dutch Language)[6] invested in the Klare Taal (Clear Language) language policy to address this issue. Klare Taal consists of a number of rules and guidelines that can be consulted by employers who aim at communicating written, oral, or visual information in a more comprehensible, consistent, and clear way. In its written form, Klare Taal is similar to the English guidelines for easy-to-read text, which we will not discuss here.[7]

We use the Klare Taal checklist[8] (see Appendix C) to investigate how the prescriptions from Huis van het Nederlands could or should be applied to the pictograph

[6] http://www.huisnederlandsbrussel.be/

[7] Similar guidelines can be found in the "Make it Simple" report on easy-to-read language (Freyhoff et al. 1998).

[8] http://www.klaretaalrendeert.be/files/Checklist%20duidelijk%20geschreven%20taal.pdf

translation case. The checklist gives us several clues on how to deal with long and/or complex input sentences, which could lead to the generation of pictograph sequences that are difficult to interpret. Note that, while the original Klare Taal checklist refers to natural language words and sentences, we apply the Klare Taal principles to the domain of pictograph translation, instead.

On the **word level**, Klare Taal recommends using short, everyday words. The use of jargon and metaphorical language should be avoided, as well as acronyms, abbreviations, and nominalisations. The Text-to-Pictograph translation system inherently solves these lexical issues by using pictograph support for words.

Unsolved issues are still found on the **sentence level**. Firstly, Klare Taal recommends using the active voice. Passive voice cannot be expressed by pictographs. In order to ensure that the reader does not mistakenly recognise the pre-verbal patient as the agent of the depicted action, it will be necessary to re-order the arguments. Secondly, the baseline translation engine does not simplify the input text, with the exception of the removal of articles and conversion into complex (Sclera) pictographs where possible. As a consequence, long input messages correspond to long pictograph translations. This practice clashes with Klare Taal's recommendation of using short sentences. To solve this issue, complex and coordinate sentences will have to be converted into multiple, independent clauses. Finally, Klare Taal advises to place the subject at the beginning of the sentence. The baseline Text-to-Pictograph translation system does not yet change the order of the pictographs and provides an almost literal translation from Dutch. To help readers familiarise themselves with the pictograph output, and to compensate for the lack of grammatical markers, the use of fixed syntactic positions will be essential.

On the levels of **text structure** and **design**, Klare Taal recommends using short paragraphs. There should be enough space between the paragraphs, and each paragraph should correspond to a new information unit. Note that, as a result of syntactic simplification, the output sequences will be much shorter, but there will also be more line breaks. While the pictographs are already rather large, their size can be adjusted, making them also an appropriate communication aid for visually impaired people. Just like different fonts should not be mixed, we opt not to combine pictographs from different sets.

Finally, with respect the **content** of the message, Klare Taal strongly recommends to take the needs of the target group into account. The pictograph sets and the Text-to-Pictograph translation system were developed with a specific group of users in mind, namely people with ID. However, we already showed that the baseline pictograph translation tool is not yet fine-tuned toward the users' real expectations and needs, and that further improvements are necessary.

Construction	Frequency	Rel. frequency
NP WW(pv,tgw,ev) NP LET()	48	2.37%
NP WW(pv,tgw,ev) NP PP LET()	43	2.12%
NP WW(pv,verl,ev) NP PP LET()	40	1.98%
NP WW(pv,verl,ev) NP LET()	33	1.63%
NP WW(pv,verl,ev) PP LET()	29	1.43%
NP WW(pv,tgw,met-t) PP LET()	25	1.23%
NP WW(pv,tgw,met-t) NP LET()	25	1.23%
NP WW(pv,tgw,met-t) NP PP LET()	21	1.04%
NP WW(pv,tgw,mv) NP PP LET()	13	0.64%
NP WW(pv,tgw,ev) AP(vrij) LET()	12	0.59%
NP WW(pv,tgw,met-t) PP PP LET()	11	0.54%
NP WW(pv,tgw,mv) NP LET()	11	0.54%
NP WW(pv,verl,ev) PP PP LET()	11	0.54%

Table 5.2: Syntactic patterns that occur ten times or more in the WAI-NOT news corpus. The relative frequency is measured with respect to the total amount of constructions (i.e., the total amount of sentences).

5.3.3 Syntactic Analysis of News Messages on WAI-NOT

Daily news items on WAI-NOT (see section 2.1) are written according to the rules of Klare Taal (see section 5.3.2), with a clear focus on brevity and clarity.[9] We analysed the full corpus of 221 WAI-NOT news messages (2,025 sentences or 16,628 words) by applying the HunPos part-of-speech tagger (Halácsy et al. 2007) and performing shallow syntactic parsing using the ShaRPa 2.0 chunker (Vandeghinste 2004). We then counted the syntactic patterns that occur ten times or more in the news corpus. The results are shown in Table 5.2. The HunPos part-of-speech tagger employs the D-Coi tagset (Van Eynde 2005).[10] Observe that all these constructions can be reduced

[9] A similar news website written in Dutch easy-to-read language is the Wablieft newspaper (http://www.wablieft.be/).

[10] NP: noun phrase; PP: prepositional phrase; AP: adjectival phrase; WW: verb; LET: punctuation; pv: finite verb; tgw: present tense; verl: past tense; ev: singular; mv: plural; met -t: second or third

to just three patterns:

- NP + Any finite verb + NP (+ PP) (234 occurrences)
- NP + Any finite verb + PP (+ PP) (76 occurrences)
- NP + Any finite verb + AP (12 occurrences)

Most sentences are short, and SVO is the most commonly used order. Note that 1,080 constructions (i.e., 53.33% of all sentences) occur only once in the entire news items corpus.

Construction	Frequency	Rel. frequency
NP WW(pv,tgw,met-t) NP PP LET()	10	0.49%
NP WW(pv,tgw,met-t) SSUB LET()	9	0.44%
NP WW(pv,tgw,met-t) PP PP LET()	8	0.40%
NP WW(pv,verl,ev) NP LET()	8	0.40%
NP RELP LET()	7	0.34%
SSUB LET()	7	0.34%
NP WW(pv,verl,ev) SSUB LET()	7	0.34%
NP WW(pv,verl,ev) NP PP LET()	7	0.34%
NP WW(pv,tgw,ev) SSUB LET()	5	0.25%
NP PP LET()	5	0.25%

Table 5.3: Syntactic patterns that occur five times or more in 2,025 randomly selected sentences from the SoNaR newspapers component. The relative frequency is measured with respect to the total amount of constructions (i.e., the total amount of sentences).

To compare the distribution of syntactic patterns across different registers, we also present the results for regular newspaper text. We randomly selected an equal amount of 2,025 sentences from the SoNaR newspapers component (Oostdijk et al. 2013), and we applied part-of-speech tagging and chunking with ShaRPa. We then counted the syntactic patterns that occur ten times or more in the corpus. Since only one result was found, we show all structures that occur five times or more. The results are presented in Table 5.3. It can be inferred that these structures are much more varied, and that - unlike the WAI-NOT constructions - some of the most commonly used constructions include a subordinate clause or a relative clause. The

person singular; SPEC: special token (such as proper nouns); NUM: numeral; VRIJ: predicative or adverbial; SSUB: subordinate clause; RELP: relative clause.

NP + Any finite verb + NP (+ PP) construction, which occurs at least 234 times in the WAI-NOT corpus, appears only 25 times in the SoNaR set. This is clear evidence for the simplicity and consistency that is pursued in the WAI-NOT syntax, as opposed to regular newspaper text.

We also analysed the internal structure of noun phrases, prepositional phrases, and adjectival phrases in the WAI-NOT news items and the SoNaR newspapers component (see Table 5.4). In WAI-NOT, for noun phrases, the most commonly appearing constructions are pronouns, nouns with a determiner, and bare nouns. Prepositional phrases consist of a preposition and a noun phrase. In the majority of cases, adjectival phrases comprise one adjective. These observations will serve as a source of inspiration for sentence compression, in particular. Results for SoNaR are similar, although the absolute frequencies are higher. This can be attributed to the fact that sentences in SoNaR are longer than easy-to-read news items in WAI-NOT.

	WAI-NOT		SoNaR	
Phrase type	**Structure**	**Frequency**	**Structure**	**Frequency**
NP	PRON	1000	N	1556
	DET N	940	PRON	1546
	N	895	DET N	1248
	DET AP N	213	DET AP N	336
	NUM N	201	AP N	239
	AP N	200	NP NUM	169
			N SPEC	127
			N N	103
PP	PREP NP	1330	PREP NP	2398
AP	ADJ	395	ADJ	568
			AP ADJ	152

Table 5.4: The internal structure of noun phrases, prepositional phrases, and adjectival phrases that occur more than 100 times in the WAI-NOT corpus and the SoNaR newspapers component.

	Phenomenon	Operations
(a)	Coordinate sentence	- Split into two or more sentences - Identify the antecedent (subject) if the subject is covert (in case of ellipsis)
(b)	Subordinate clause	- Detach from the main clause
(c)	Clause order	- Determine the subordinate clause's temporal or logical position with respect to the main clause
(d)	Participial phrase	- Detach from the main clause - Identify the antecedent (subject)
(e)	*(om) te* 'to' + infinitive clause	- Detach from the main clause - Identify the antecedent (subject)
(f)	Relative clause	- Detach from the main clause - Identify the antecedent (subject, direct object, or indirect object)
(g)	Embedded appositive clause	- Place the verb *zijn* 'to be' in front of the appositive clause - Identify the antecedent (subject)
(h)	Adverbial phrase or prepositional phrase in theme position	- Move to the back of the sequence
(i)	Covert subjects or objects in the newly created independent clause	- Place all the antecedents at their appropriate positions within the newly created independent clause
(j)	Non-subject-verb-object order	- Convert to subject-verb-object order - Cluster all the verbs at the verb position
(k)	Passive voice	- Swap the agent and the patient - Remove *worden* 'to be'
(l)	*Optional operation: compression*	*- Delete pictographs*

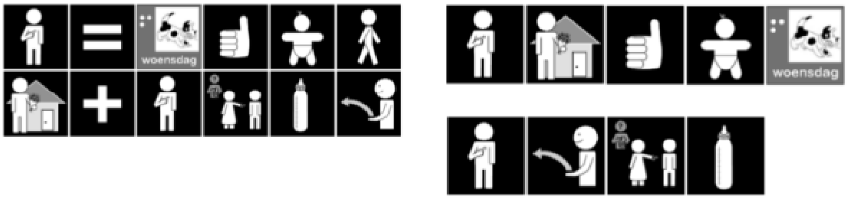
Table 5.5: Syntactic phenomena to be treated by the syntactic simplification module for pictograph translation.

5.4 Phenomena to be Treated by the Syntactic Simplification Module for Pictograph Translation

Based on the previously discussed guidelines and sources of inspiration, we identified a number of "complex" syntactic phenomena for the simplification system to deal with. Note that the system is not developed for generating textual output, but for generating pictograph output, with simplified text as an intermediary step. Table 5.5 summarises the syntactic phenomena to be treated. We present a number of examples below. The parenthesised letters refer to the identifiers of the operations in Table 5.5. For brevity, only Sclera examples are shown.

Pictograph sequences should **not be too lengthy**. This can be achieved by splitting coordinate sentences into multiple, shorter clauses **(a)** (see Figure 5.9).

Non-simplified | **Simplified**



Ik ben woensdag een lieve baby gaan bezoeken en ik heb hem een papfles gegeven. 'On Wednesday I went to visit a cute baby and I gave him the bottle.'

Read as: "I visited a cute baby on Wednesday. I gave him the bottle."

Figure 5.9: Splitting coordinate sentences into multiple, shorter clauses.

The presence of subordinate clauses **(b)** (followed by step **(c)**), participial phrases **(d)**, *(om) te* 'to' + infinitive clauses **(e)**, relative clauses **(f)**, and appositive clauses **(g)** may hamper the user's understanding of the pictograph message. We **split complex sentences** into multiple, shorter clauses. Each of these clauses is transformed into a simple, independent sentence. To achieve this, we must **identify the antecedents** of any unrealised arguments that the newly created sentence may contain, and insert them at their appropriate positions **(i)** (see Figure 5.10, where the complex sentence is split, and the antecedent *oma* 'grandmother' is retrieved).

The translation system works on the sentence level. While global sentence ordering on the message level is beyond the scope of this work, we will still consider the **temporal and logical ordering of individual clauses (c)**.[11] As subordinating conjunctions, for which no pictographs are available, are deleted during translation, the

[11] While different interpropositional relations may hold between two sections of a text, their identification does not necessarily depend on grammatical signals. Instead, recognition of a relation is often made on an entirely semantic or functional basis. While extensive theoretical work has been de-

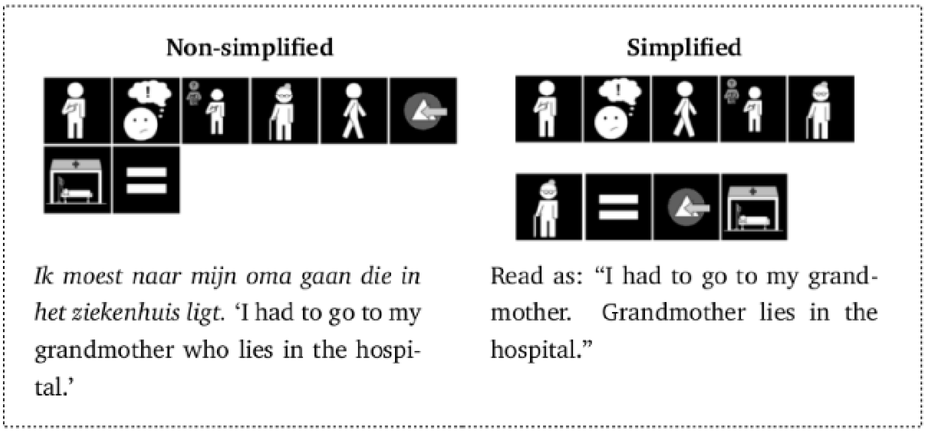


Figure 5.10: Splitting complex sentences into multiple, shorter clauses.

(chrono)logical placement of the clauses will aid the reader in better understanding the order of the events described (see Figure 5.11, where temporal re-ordering took place).

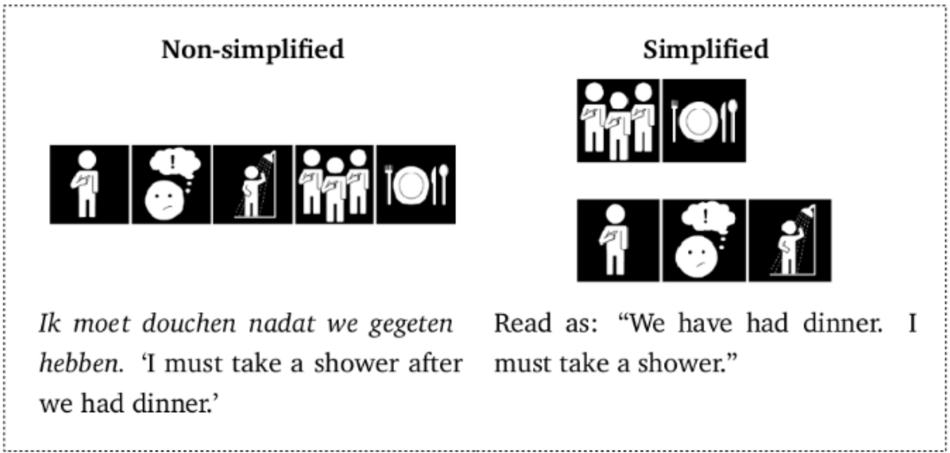


Figure 5.11: Temporal re-ordering of the clauses.

Word order in Dutch varies across clause types. We consistently display pictograph sequences as **SVO-type sequences (j)**, with a complete verb group located at the V position, and **convert passive constructions into active ones (k)** (see Figure 5.12, where the original subordinate clause is converted into an SVO-type clause,

voted to this topic (see, for example, the Rhetorical Structure Theory by Thomas (1995)), we have no knowledge of natural language processing tools for Dutch that are able to infer these types of relations between sections of a text automatically. Therefore, for the time being, the re-ordering module relies solely on the explicit presence of temporal and logical indicators.

and the passive construction is transformed into an active one). Fixed pictograph order compensates for the loss of function words or grammatical markers.[12] Adverbs and prepositional phrases located in front of the subject are detopicalised **(h)**.

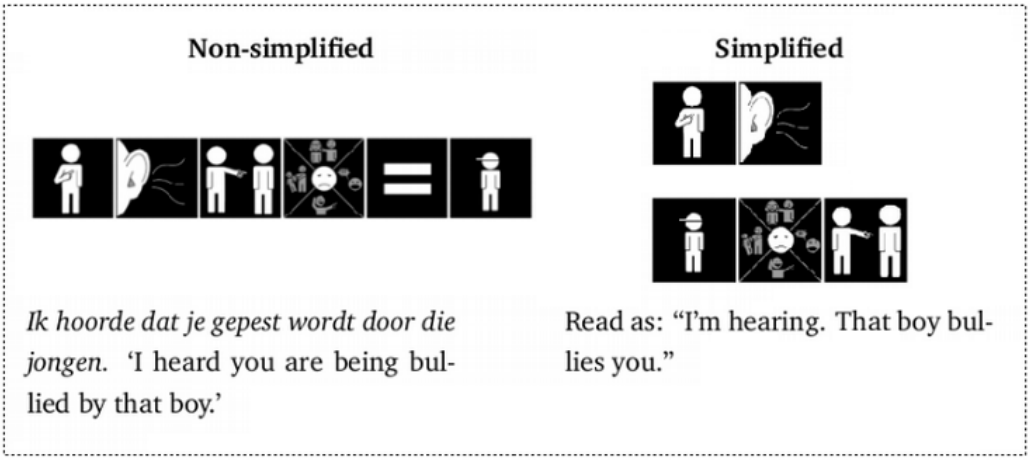


Figure 5.12: Conversion into an SVO-type clause and transformation into active voice.

5.5 System Description

In this section, we describe the syntactic simplification module. The module consists of three steps: pre-processing (section 5.5.1), syntactic analysis with Alpino (section 5.5.2), and the actual simplification operations (section 5.5.3). We then present the results of our evaluations (section 5.5.4).

5.5.1 Pre-processing

In the pre-processing step, all quotation marks and emoji are removed, as they are not converted into pictographs. The input may consist of one or more sentences. As the syntactic parser works on the sentence level, the sentences are split based on punctuation marks (period, question mark, semicolon, and exclamation mark).

5.5.2 Applying Alpino

Alpino (van Noord 2006) is a syntactic parser of Dutch which aims at full, accurate parsing of unrestricted text. Alpino incorporates knowledge-based techniques, such

[12] Our choice for SVO is motivated by the fact that it is the standard word order in main clauses.

as a Head-Driven Phrase Structure (HPSG) grammar and a lexicon (Pollard & Sag 1994), and corpus-based techniques. Based on the categories that are assigned to words and word sequences, and a set of grammar rules compiled from the HPSG grammar, Alpino returns the best analysis for a given input sentence.[13] The parse is rooted by an instance of the top category. If there is no parse covering the complete input, the parser finds all parses for each substring.

We run Alpino on the input message and generate the full parses of all sentences (see Example 5.1 and Figure 5.13). If parsing takes too long, a time-out will occur, and shallow linguistic analysis (part-of-speech tagging and lemmatisation, see section 3.1) is performed instead.

(5.1) Ik ken jou niet en wil geen mailtjes meer krijgen van jou.
I know you not and want no emails anymore receive of you.
'I don't know you and do not want to receive any more emails from you.'

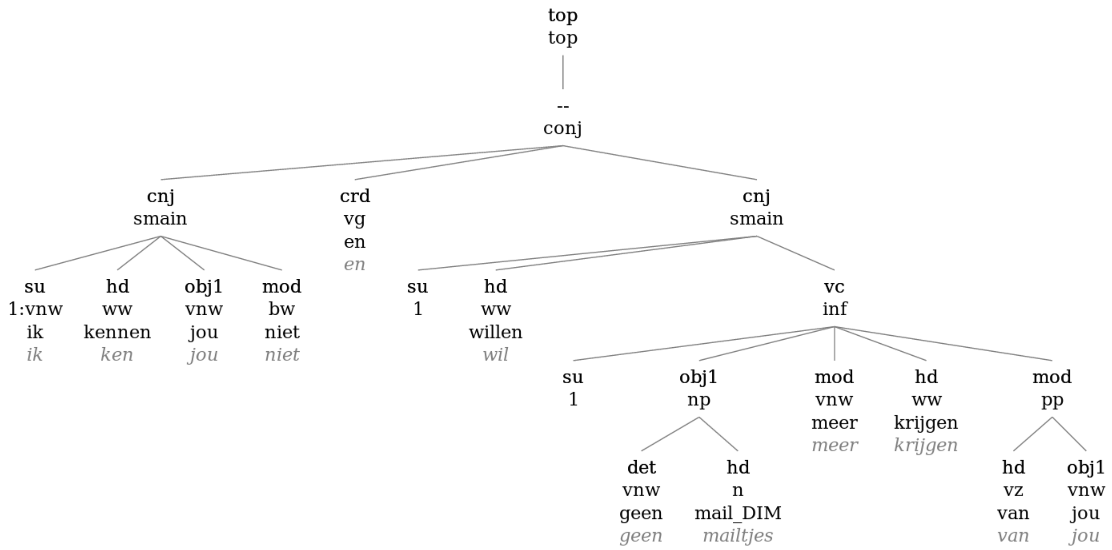


Figure 5.13: Alpino parse of the coordinate sentence that is shown in Example 5.1.

5.5.3 Syntactic Simplification

If the parse succeeds, it is time for the simplification module to start processing Alpino's output. In our approach, each *clause* from the original input message is

[13]We use the *-veryfast* option.

converted into a simple *sentence*, i.e., a sentence with only one main (independent) clause. A sentence is eventually translated into a pictograph sequence, and followed by a line break.

This module replaces the shallow linguistic analysis phase of the baseline system.

The syntactic simplification for pictograph translation consists of two steps. First (section 5.5.3.1), the Alpino parse is processed, the clauses are identified, and the new sentences are created (operations **(a)** to **(g)**). At this stage, the module also detects interrogation and passivity, and retrieves the head and the grammatical function of the antecedents, if applicable. The latter operation is necessary in the case of ellipsis, appositives, relative clauses, and *(om) te* 'to' + infinitive clauses, where either the subject or one of the objects is covert or unrealised and must be retrieved in order to enable the creation of a simple sentence. The second step (section 5.5.3.2) consists of re-ordering the syntactic constituents and placing the antecedents and verbs at their appropriate positions within the newly created sentences, with a primary aim of creating active SVO-type sentences (operations **(h)** to **(k)**). We also shortly describe a basic module for sentence compression (section 5.5.3.3).

Note that the order of the simplification operations is not chosen arbitrarily. One sentence may contain several phenomena that could be simplified. Candido Jr. et al. (2009) apply their operations in cascade. At each iteration, the system verifies the various phenomena to be simplified in order. When a phenomenon is identified, its simplification is executed. The resulting simplified sentence is re-parsed and goes through a new iteration. Re-parsing slows down the simplification system considerably. In our system, the input sentences are only parsed once. Processing Alpino's output by means of recursion and loops allow us to deal with all syntactic phenomena to be simplified in an elegant way. We will illustrate this principle by means of an elaborate example in section 5.5.3.4.

The order of the operations that we apply to the output of the parser is shown in Table 5.5. This order was rationally, based on the simple idea that some operations are needed as a prerequisite to make other operations work. For instance, for the conversion of passive voice into active voice, the theme and the agent, if any, are swapped (and *worden* 'to be' is removed). In most cases, this operation is self-evident. However, if the passive construction would happen to be located inside a relative clause, it is necessary to identify the antecedent of the unrealised patient before this transformation can take place. Therefore, the identification of the antecedents should occur before the conversion of passive constructions into active ones.

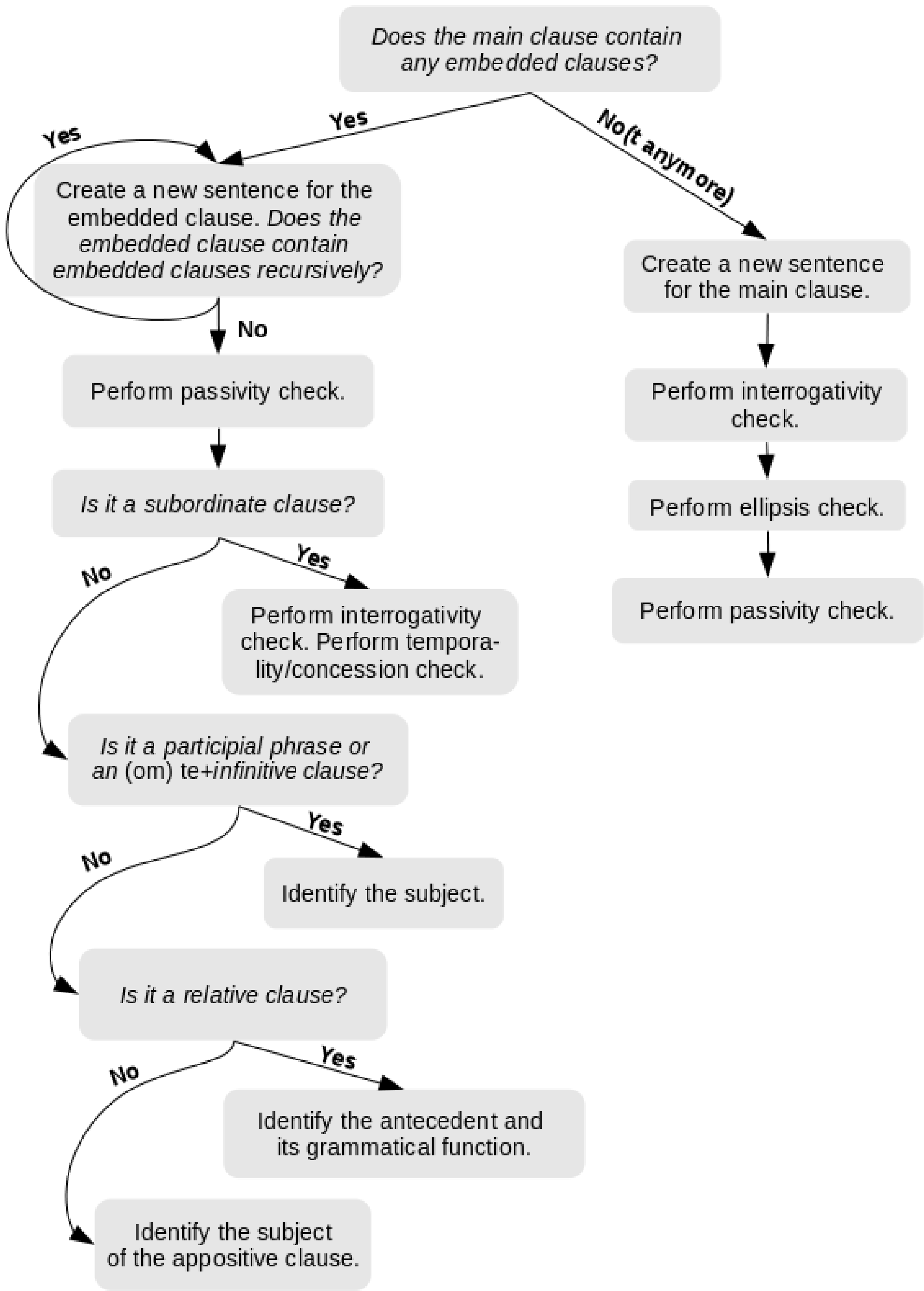


Figure 5.14: Schematic representation of the first step (creating sentences and identifying syntactic phenomena) in the syntactic simplification procedure.

5.5.3.1 Step One: Creating Sentences and Identifying Syntactic Phenomena

In this step, complex and coordinate sentences are split into multiple, shorter sentences (originally corresponding to clauses), and target structures to be simplified are identified. The pseudocode for this step is presented in Appendix D.1. The line numbers used in this section correspond to those that are used in the pseudocode. A schematic representation of this step is given in Figure 5.14.

5.5.3.1.1 Main Clauses The simplification system creates a new sentence for every type of main clause (*SMAIN*,[14] *DP*,[15] or *SV1*[16]) in the Alpino parse (operation **(a)**) (l.[48-63]). A sentence is represented as a multiple-level hierarchy of phrases and words, as can be seen in Figure 5.15. Phrases receive two features: grammatical category and function. For words, we retain five features: token, lemma, part-of-speech tag, function, and index (if applicable). Whereas the original Text-to-Pictograph translation tool would convert the input message into a flat array of words, linguistic analysis is now no longer shallow.

Main clauses may stand alone (and when they do, they are the same as a simple sentence), or they may be joined to other clauses, such as subordinate clauses, relative clauses, participial phrases, *(om) te* ‘to’ + infinitive clauses, and appositives. For now, these clauses are removed from the main clause. New sentences will be created for them (l.[4- 47]), as will be shown in the following sections.

The Alpino parser performs separable verb detection (see section 3.1). If a separable verb is found, we remove the separable particle and create a compound token. An input message such as *Tim nam alle koekjes weg* ‘Tim took all the cookies away’ would then become *Tim weg-nam alle koekjes* ‘Tim took-away all the cookies’. Note that this sequence does not correspond to a grammatical sentence in Dutch. However, this is not important for pictograph translation, since only the lemma is needed for translating the message.

A number of syntactic properties are identified by the system and new sentence-level features are created for different types of main clauses. Note that these features are re-used in the second phase of the simplification process (see section 5.5.3.2).

[14] Main clause.
[15] Discourse part.
[16] Verb-initial clause, such as interrogative sentences.

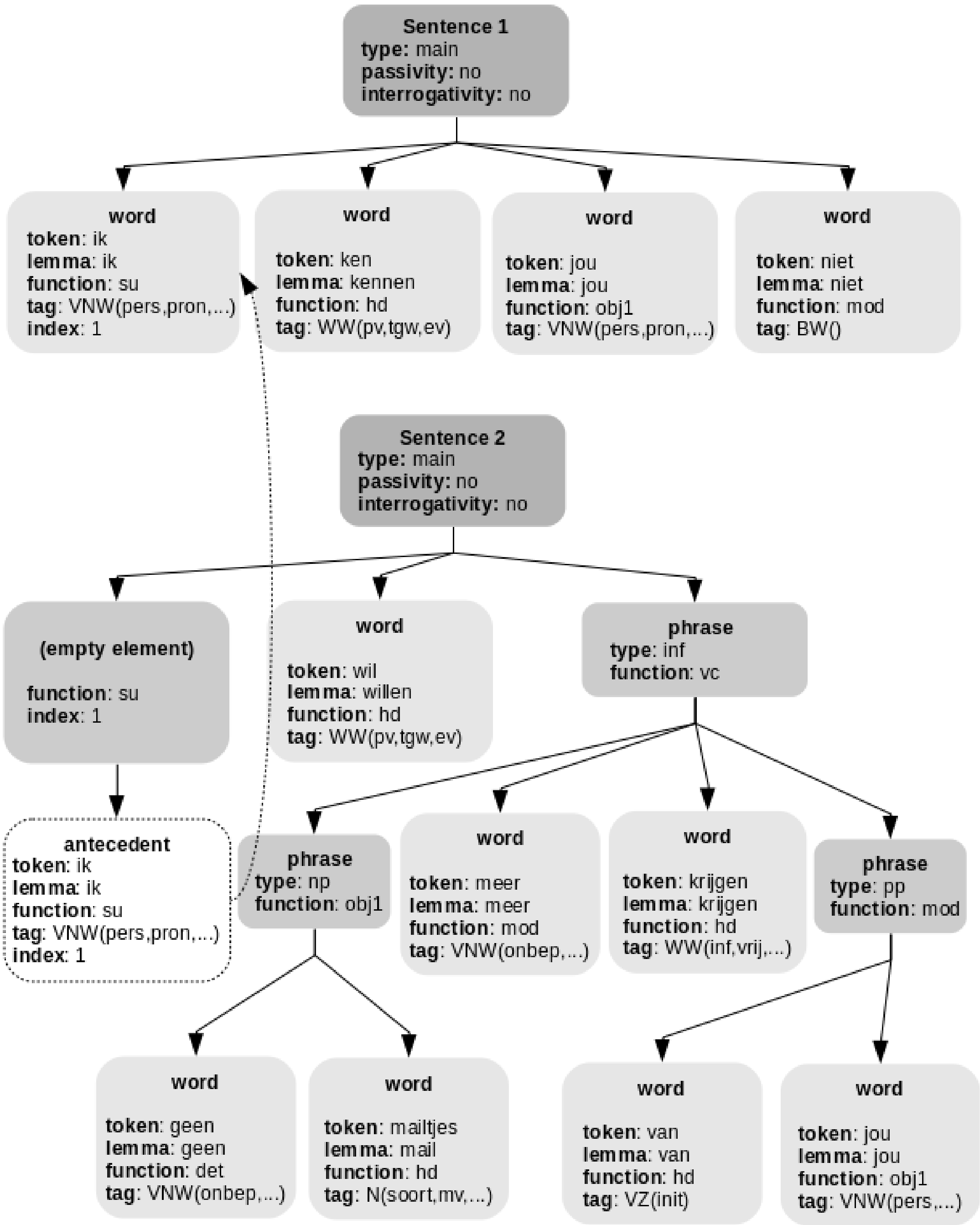


Figure 5.15: The Alpino parse from Figure 5.13 after splitting, and antecedent detection of the unrealised element. The empty element of the second main clause shares its index with the subject of the first main clause. This way, the antecedent can be identified.

- If the main clause begins with a question word, we mark it as an *interrogative* sentence. The same goes for the verb-initial clauses (SV1-type clauses) (l.[50-52]).[17]

- If the main clause is missing an overt subject, for instance, as a result of coordination with ellipsis, the original subject is identified (l.[53-60]). An example of this is shown in Figure 5.13, where the subject, *I*, is expressed only once. When splitting this sentence, however, the subject must be repeated - else, the predicate would be missing one of its arguments: *ik ken jou niet* 'I don't know you' and *ik wil geen mailtjes meer krijgen van jou* 'I do not want to receive any more messages from you' (see Figure 5.15). The Alpino parser marks covert constituents by means of an unrealised subject element with an index. This index connects the unrealised element to a noun, pronoun, or noun phrase in another clause, allowing us to identify the subject. If the subject is a noun phrase, we do not retrieve the whole phrase, but only the head noun. At this point, the newly retrieved subject is added as a sentence-level feature. It will be inserted at its appropriate position in the second phase.[18]

- If a passive construction is detected by the Alpino parser, the sentence receives the *passive* feature (l.[61-63]).

5.5.3.1.2 Subordinate Clauses, Relative Clauses, Participial Phrases, and *(Om) te* + Inf. Clauses For every main clause (*SMAIN*, *DP*, or *SV1*), the simplification system verifies whether it contains any subordinate clauses, relative clauses, participial phrases, or *(om) te* 'to' + infinitive clauses (l.[4-47]). If this is the case, the simplification module creates a new simple sentence for each of these clauses (operations **(b)**, **(d)**, **(e)**, and **(f)**). Note that the simplification system also looks for subordinate clauses and any other clauses that may appear within these newly created sentences recursively until all clauses are retrieved.[19]

A number of syntactic properties are identified by the system and new sentence-level features are created.

---

[17]Unless the verb's *stype* value equals *imperative*.

[18]Looking at the parse in Figure 5.13, it can be seen that the covert subject is repeated as an unrealised element with an index at different levels within the hierarchy, serving as the argument of multiple predicates. As shown in Figure 5.15, we only retain the subject of the highest-level predicate, since only one subject pictograph is needed for translation.

[19]This is a form of depth-first search.

- If a passive construction is detected by the Alpino parser, the sentence receives the *passive* feature (l.[11-13]).
- If the subordinate clause begins with a question word, we mark it as an *interrogative* sentence (l.[16-18]).
- Subordinate conjunctions provide a transition between the main clause and the subordinate clause. This transition could be, for instance, a temporal or a concessive relationship. We made a list of Dutch subordinate conjunctions that express concession (*hoewel* 'although', *ofschoon* 'although', *al* 'although') or temporal priority/anteriority (*voordat* 'before', *voor* 'before', *na* 'after', *nadat* 'after') with respect to the events in the main clause.[20] If the simplification system detects a subordinate clause that is headed by any of these conjunctions, we move that clause in front of or behind the main clause, depending on the type of relation expressed (operation **(c)**) (l.[19-22]). Since subordinate conjunctions, for which no pictographs exist, are deleted from the pictograph sequence, the (chrono)logical placement of the newly created sentences can help the reader understand the temporal (*X happens before Y*) or logical (*X causes Y*) order of the events depicted.
- If the clause is a participial phrase or an *(om) te* 'to' + infinite clause, its subject is marked by Alpino as an unrealised element with an index (l.[23-29]). This index allows us to identify the antecedent of the participial phrase in the main clause. Once the subject has been found, it is added as a new sentence-level feature. The subject is inserted at its appropriate position in the second phase.
- The relative clause's antecedent, which appears in the main clause, could be a noun, a pronoun, or a noun phrase, and can be identified by means of an index (l.[30-37]). Since the relative clause's antecedent may fulfill multiple grammatical functions within the relative clause, such as the function of subject or (in)direct object, we also retrieve the unrealised element's grammatical function. Note that many of the previously discussed rule-based simplification systems (see section 5.2) assume that a relative clause's antecedent must typically fulfill the subject role. This is not always the case, as we will show later on in Figure 5.17. For now, the antecedent and its grammatical function are

[20] This list might not be exhaustive and can be extended.

added as new sentence-level features. The antecedent will be inserted at its appropriate syntactic position in the second phase.

5.5.3.1.3 Appositive Clauses An appositive clause is defined as an arrangement of words in which a noun or noun phrase is followed by another noun or noun phrase that refers to the same person or object.

The simplification system searches for the antecedent of the clause that refers to the same person or object (operation **(g)**) (l.[38-44]). As mentioned before, this antecedent can be a noun or a noun phrase. The antecedent is added as a new sentence-level feature and will be re-used in the second stage of the simplification procedure (see section 5.5.3.2).

5.5.3.2 Step Two: Changing the Order of the Constituents

The second step mainly consists of re-ordering the syntactic constituents and placing the antecedents at their correct[21] syntactic positions (subject or object position) within the newly created sentences. This process consists of several steps, which are applied in order. The pseudocode for this step is presented in Appendix D.2. The line numbers used in this section correspond to those that are used in the pseudocode. A schematic representation of this step is given in Figure 5.16.

- Prepositional phrases and adverbs that are located at the beginning of the sentence, in front of the subject, are detopicalised and moved to the back of the sentence (operation **(h)**). Examples of this are *gisteren zijn we gaan zwemmen* 'yesterday we went swimming' or *volgende week word ik 15 jaar* 'next week, I'm turning 15' (l.[2-4]).

- The simplification system detects words that indicate a negative polarity, such as *niet* 'not' and *geen* 'no'. When such a word is found, we look for its head within a window size of three words, i.e., we look for a verb, adjective, or adverb in the three preceding and the three following words. When the head is found, we add the value *negative* to its polarity feature. With this operation, we allow the Text-to-Pictograph translation tool to display the negation pictograph right next to the negated pictograph (l.[5-7]).

[21] According to our SVO-type pictograph grammar.

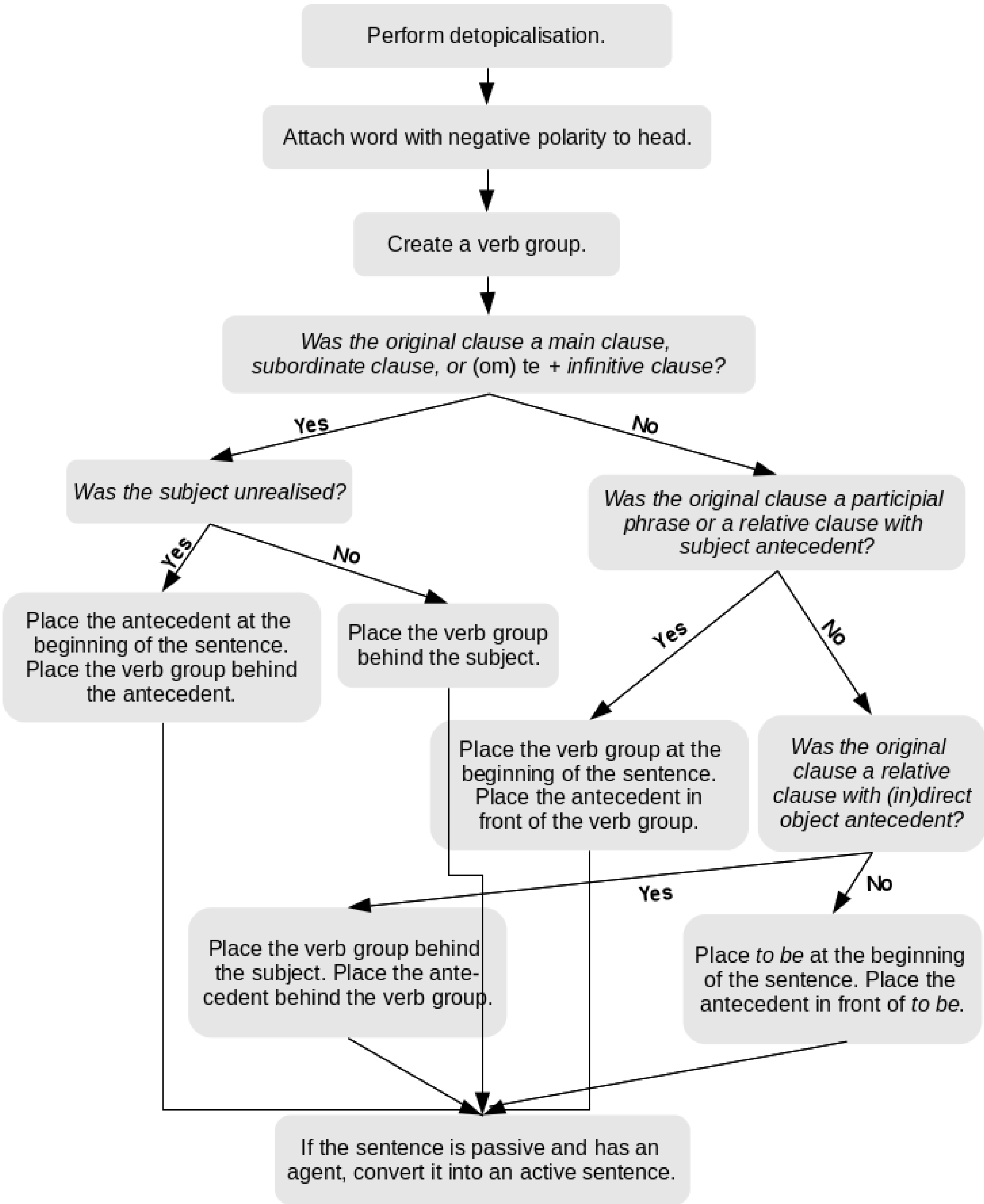


Figure 5.16: Schematic representation of the second step (changing the order of the constituents) in the syntactic simplification procedure.

- The system retrieves all verbs that occur within the sentence, removes them from their original positions, and clusters them into a verb group (l.[8]). In Example 5.1, for instance, *wil* 'want' and *krijgen* 'receive' would be removed from the second main clause, and temporarily stored into a newly created verb group. Note that the verb group is also created if only one verb is found.

- At this point, the antecedents that were retrieved in the identification step (see section 5.5.3.1), as well as the newly created verb group, must be inserted at their appropriate syntactic positions, with the purpose of creating SVO-type sentences (operations **(i)** and **(j)**):

  - For sentences that originate from main clauses, subordinate clauses, interrogative clauses, and *(om) te* 'to' + infinitive clauses, the system first checks whether a subject antecedent was retrieved. If this is the case, the system places the antecedent at the beginning of the sentence (result: SO). The verb group is inserted behind the subject (result: SVO). Note that interrogative clauses are also converted into SVO-type clauses. Interrogation is marked by a question word pictograph at the beginning of the sentence and/or a question mark pictograph at the end of the sentence (l.[9-18]).
  - For sentences that originate from participial phrases and relative clauses with a subject-type antecedent, the system first places the verb group at the very beginning of the sentence (result: VO), and then puts the antecedent in front of that verb group (result: SVO) (l.[19-21]).
  - For appositives, the system first adds *zijn* 'to be' to the beginning of the sentence, and then places the antecedent in front of *zijn* 'to be' (l.[25-27]).
  - For sentences that originate from relative clauses with a direct object or indirect object antecedent, the system first moves the verb group behind the subject (result: SV), and then places the antecedent behind that verb group (result: SVO). An example of a relative clause with a direct object (see Example 5.2) is given in Figure 5.17, where the direct object antecedent *stuur* 'steering wheel' and the verb *vastmaak* 'attach' are inserted at their appropriate positions (l.[22-24]).

(5.2) (Ik heb een stuur dat) ik aan mijn rolstoel vastmaak.
(I have a steering-wheel that) I to my wheelchair attach.
'(I have a steering wheel that) I can attach to my wheelchair.'

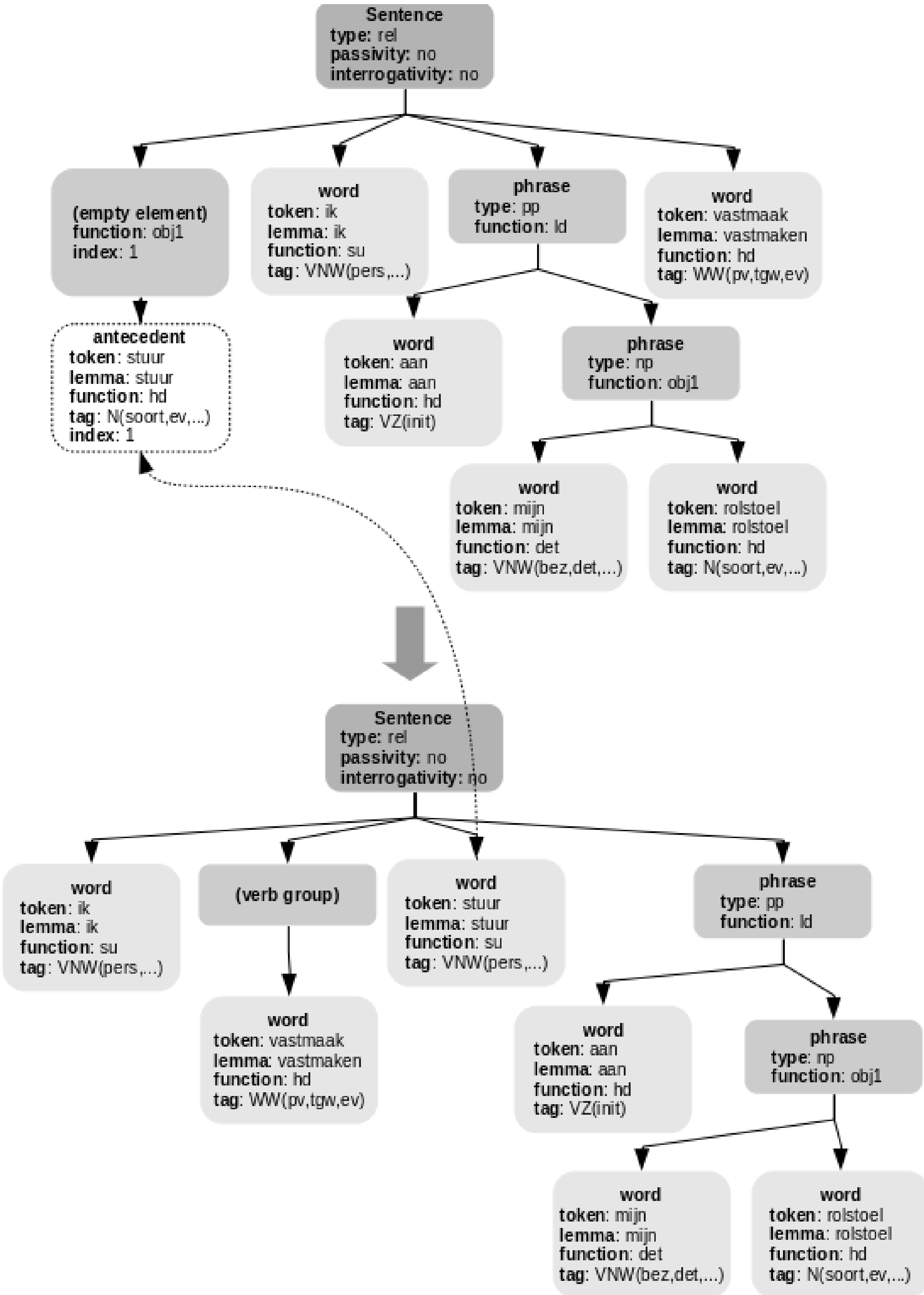


Figure 5.17: Re-ordering of the relative clause in Example 5.2. The object antecedent *stuur* 'steering wheel' and the verb *vastmaak* 'attach' are inserted at their appropriate syntactic positions. The new sequence can be read as *I attach the steering wheel to my wheelchair*.

- For sentences that carry the *passive* feature, we first check if the agent is overtly expressed by means of a prepositional phrase introduced by the preposition *door* 'by'. If this is the case, the agent is moved to the beginning of the sentence, and the patient is moved behind the verb phrase (operation **(k)**). The verb *worden* 'to be' is removed from the sequence (l.[29- 35]). If the agent is not expressed, we simply leave the sentence as such.

5.5.3.3 Optional: Compression

The compression module is an optional operation that acts on the output of the syntactic simplification module. It displays all nouns, pronouns, verbs, and numbers, but removes all function words (articles, prepositions, etc.), adverbs, and adjectives from the pictograph sequence. Our choice to remove function words originates from the observation that these pictographs, if any are available, are typically rather abstract and difficult for new pictograph users. We can remove these pictographs from the pictograph sequence without drastically changing the content of the message. We delete adverbs and adjectives (unless they are subject complements), because the results of our shallow syntactic analysis of the WAI-NOT news message corpus (see section 5.3.3), which contains messages that are addressed at people with ID, suggest that the absence of these part-of-speech categories is far more common than their presence. When a user becomes more familiar with the Text-to-Pictograph translation tool, a caregiver can decide to disable the compression module, and let the user work with the regular simplification module instead.[22]

5.5.3.4 Elaborate Example: Multiple Simplification Operations

We illustrate a complete simplification procedure by means of an elaborate example (see Figure 5.18). Note that this specific example was made up for the purpose of illustrating various complex syntactic phenomena at once.

The simplification system identifies the main clause and is able to detect a subordinate clause and a relative clause. The subordinate clause and the relative clause are detached from the main clause in operations **(b)** and **(f)**. The system checks for

[22] In our user study on syntactic simplification (see section 5.5.4.3), 20 out of 28 participants indicated that the compression module would be helpful for their target user.

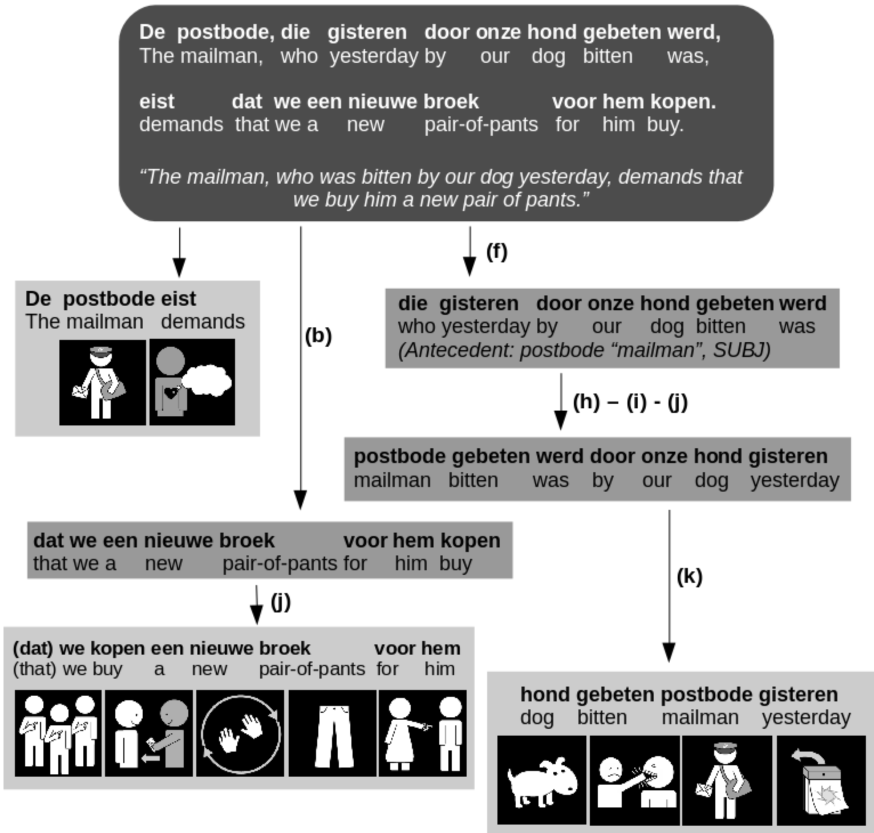


Figure 5.18: Elaborate example of the simplification procedure.

more embedded clauses recursively, but none are found. The splitting operations reveal that the simplification system will leave the pictograph user with three sentences (or pictograph sequences). Interrogativity and passivity are checked for every newly created sentence. Note that the relative clause's antecedent *postbode* 'mailman' is identified, as well as its grammatical function. In this case, *postbode* 'mailman' would be the grammatical subject of the relative clause. This ends the first step.

The second step is mainly concerned with re-ordering. The main clause is not interrogative, nor passive. Its constituents must not be re-ordered, as it is an SV(O)-type clause. No further simplification operations are needed.

The system moves on to the subordinate clause. It is not interrogative, nor passive. However, its current order is SOV. Operation **(j)** takes care of the re-ordering. Furthermore, the subordinate conjunction is removed, as it cannot be translated into a pictograph.

The relative clause is the most challenging one. The system first detopicalises the adverb *gisteren* 'yesterday' by moving it to the back of the sentence **(h)**. The antecedent *postbode* 'mailman' is placed at the beginning of the sentence, in front of the verb, as it is a subject-type antecedent **(i)**. After this step, the word order is SOV. Operation **(j)** takes care of the re-ordering, making sure that the verb group, which consists of an auxiliary and a past participle, is moved to the V position. One more simplification operation can be applied **(k)**. The relative clause carries the *passive* feature, and an agent (*hond* 'dog') is identified. The system moves the agent to the beginning of the sentence, and puts the patient behind the verb phrase. The verb *worden* 'to be' is removed.

The resulting pictograph sequences can be read as *the mailman has a demand, we have to buy him a new pair of pants*, and *our dog bit the mailman yesterday*.[23]

5.5.4 Evaluation

Section 5.5.4.1 presents the results of an automated evaluation on newspaper text and WAI-NOT messages, using different evaluation metrics to compare the system's output to a reference translation. In section 5.5.4.2, we describe a manual evaluation. We count the amount of syntactic phenomena to be simplified and calculate the system's precision and recall on newspaper text and the WAI-NOT test set. Finally, we present the results of the user studies in section 5.5.4.3.

5.5.4.1 Automated Evaluation

We simplify 100 sentences (1,942 words – an average of 19.42 words per sentence) from De Standaard,[24] a Belgian newspaper, and we post-edit the output according to the simplification operations that we set out earlier (see Table 5.5). News articles in De Standaard present a considerable number of complex syntactic phenomena, allowing us to put the simplification pre-processing module to the test. We also simplify and post-edit our test set of 50 WAI-NOT messages (see Appendix B.2).

We compare the reference simplification to the output of the simplification system using **BLEU** (Papineni et al. 2002), **NIST** (Doddington 2002), **Word-Error Rate** (WER), **Position-Independent Word Error Rate** (PER), **Human-Targeted Transla-**

[23]Note that one possible extension for the clause re-ordering operation described in section 5.5.3.2 would be to include temporal adverbs like *gisteren* 'yesterday', instead of just subordinate conjunctions that express temporality.

[24]http://www.standaard.be/

tion Error Rate (Snover et al. 2006), and **ROUGE** (Lin 2004) scores (see Table 5.6 and Table 5.7).[25] ROUGE is a set of metrics for evaluating automatic summarisation of texts. ROUGE-1 calculates the overlap of unigrams between the system and reference summaries, whereas ROUGE-2 measures the overlap of bigrams.[26] The table shows the F-score for both ROUGE metrics. Significance levels are calculated for BLEU and NIST. The lower bound condition is the one that does not syntactically analyse, nor simplify the input text. The simplification condition uses the complete simplification module.

System	BLEU↑	NIST↑	WER↓	PER↓	HTER↓	ROUGE-1↑	ROUGE-2↑
Lower bound	31.87	6.76	53.06	19.32	35.39	87.53	44.89
Simplification	94.96**	10.74**	2.41	1.35	2.00	98.95	96.12

Table 5.6: Evaluation of the syntactic simplification module on 100 sentences from De Standaard. $*p < 0.05, **p < 0.01$.

System	BLEU↑	NIST↑	WER↓	PER↓	HTER↓	ROUGE-1↑	ROUGE-2↑
Lower bound	20.70	5.66	20.61	50.81	88.36	88.36	50.50
Simplification	87.44**	9.56**	3.36	6.75	97.53	97.53	94.71

Table 5.7: Evaluation of the syntactic simplification module on 50 WAI-NOT messages. $*p < 0.05, **p < 0.01$.

5.5.4.2 Manual Evaluation

Fine-grained, manually calculated results for De Standaard are presented in Table 5.8. Note that the system does not oversimplify on this test set: It does not perform any unnecessary simplification operations. These results reveal that, relatively speaking, most errors are made during the antecedent identification and insertion steps. We dug deeper and observed that these errors are a consequence of simplifying a syntactic parse that was erroneous in the first place.[27] For instance, relative clause at-

[25]Evaluation metrics for the specific task of evaluating simplification output exist, but their focus lies on lexical simplification and readability, rather than syntactic simplification. One such example is SARI (Xu et al. 2016). SARI correlates better with human evaluations than BLEU when multiple reference simplifications are available.

[26]The difference between the ROUGE-n precision and BLEU is that BLEU introduces a brevity penalty term, and also computes the n-gram match for several sizes of n-grams.

[27]We refer to van Noord et al. (2013) for an evaluation of the Alpino parser.

		De Standaard		WAI-NOT	
		Cases	**Correct**	**Cases**	**Correct**
(a)	Split coordinate sentences	37	36	97	90
(b)	Detach subordinate clauses	51	50	18	17
(c)	Change clause order (when temporality or a logic relation is expressed)	1	1	1	1
(d)	Detach participial phrases	0	0	3	3
(e)	Detach *(om) te* 'to' + infinitive clauses	25	25	0	0
(f)	Detach relative clauses	31	31	0	0
(g)	Detach appositives	16	15	0	0
(h)	Move adverbial clauses or Prepositional Phrases in theme position to the back	27	26	4	4
(i)	Insert any covert subjects or objects in the newly created independent clause, at the correct position	89	73	3	3
(j)	Convert non subject-verb-object order into subject-verb-object order	153	147	49	49
(k)	Convert passive voice into active voice	9	9	1	1
	Total	**439**	**413**	**176**	**168**
	Recall		100%		100%
	Precision		94.10%		95.5%
	F-score		**96.9%**		**97.7%**

Table 5.8: Analysis of the complex syntactic phenomena that occur within the 100 sentences from De Standaard and 50 messages from WAI-NOT.

tachment can be ambiguous, and the parser's erroneous decisions are propagated. Nevertheless, the results look promising. In particular, non-SVO order occurs quite a lot, but the system manages to deal appropriately with this phenomenon in most cases. Additionally, nearly all complex and coordinate sentences are split correctly.

It can be seen that WAI-NOT emails differ from newspaper text in various ways (see Table 5.8). Firstly, relatively speaking, WAI-NOT messages contain a much larger amount of coordinate sentences. We found that most long messages written by WAI-NOT users of consist of multiple simple sentences that are chained together without punctuation signs or conjunctions, due to the users' limited writing skills. In most cases, Alpino is able to successfully convert these messages into multiple, fully grammatical discourse units. Nevertheless, the user's input is not always grammatical, and ungrammatical input sometimes forms an obstacle for the parser. Furthermore, there is a relatively small amount of complex phenomena. This can be attributed to the fact that people with low literacy skills experience problems with all kinds of advanced syntactic constructions. However, we can still expect these constructions to appear on social media websites, where the user will be able to read his/her family members' status updates and subscribe to fan pages. Finally, the presence of non-SVO order is primarily caused by inversion in interrogative sentences.

5.5.4.3 User Studies

For the human evaluation of the syntactic simplification tool for pictograph translation, we have decided to collaborate with the target users' environment, i.e., their family and teachers, speech therapists, or caregivers. The reasons for this are the following:

- Family members, teachers, etc. play an important role in the communication process with the target users.
- Participants are required to answer 47 questions (including optional questions). Therefore, participants must concentrate for a prolonged period of time.
- Meta-questions concerning the difficulty of the pictograph translations may be too difficult to answer for some people who have limited expressive and/or receptive skills.

We ask the participants to keep one specific target user in mind while completing the survey, for instance, one of their students, or their son/daughter.

The first part of the questionnaire inquires the respondent about a number of characteristics of the user, such as his/her age, receptive communication skills, expressive communication skills, social media use, and the difficulties that he/she experiences when using email or social media websites.

We then present the respondent with 12 pictograph messages, which are the result of automated translation from Dutch text into Sclera or Beta pictographs. All input messages are taken from the WAI-NOT corpus. The 12 messages cover a total of six different syntactic phenomena: passive constructions, (long) coordinate sentences, subordinate clauses, relative clauses, appositive clauses, and subordinate clauses containing a passive construction.

For each of these six phenomena, we show the participant two different pictograph messages: one sentence that is translated into pictographs by means of the baseline Text-to-Pictograph translation system, and another sentence (of similar length) that is translated into pictographs by means of the new system that makes use of the syntactic simplification pre-processing module. All 12 pictograph messages are presented to the respondents in random order.[28] We do not inform the participants of the objective of the task at hand (i.e., determining whether simplified messages are easier to decipher and considered "less difficult" than non-simplified messages). The meaning (lemma) of each individual pictograph is shown, as we are interested in the combined semantics of the complete message, rather than the meaning of the individual pictographs. For each pictograph message, we ask the participant to answer the following questions:

- Wat betekent deze pictozin volgens u?[29]
- Zou de gebruiker de betekenis van deze pictozin kunnen achterhalen?[30] *(A 10-point scale ranging from "certainly not" to "certainly" is shown.)*
- (Optioneel:) Indien de gebruiker de betekenis waarschijnlijk niet (goed) zou kunnen achterhalen, wat zijn volgens u de redenen hiervoor? [31]

The objective of this experiment is to verify whether the respondents provide more accurate translations for the simplified messages than for the non-simplified messages, and whether their subjective difficulty ratings and comments reflect the added value of the syntactic simplification module.

[28] Another possibility would have been to present participants with simplified and non-simplified translations of the *same* sentence, instead of *different* sentences of similar length (that present the same syntactic phenomenon). However, by doing this, participants would have been more likely to find a connection between the six sentence pairs, which could lead to a priming effect: The exposure to a stimulus could activate a concept in memory that is then given increased weight in subsequent judgment tasks. In order to reduce bias, this experiment could have been improved upon by creating two different versions of the survey: one version showing either a simplified or a non-simplified translation of each of the 12 sentences, and another version showing the other translations. Seeing that we were able to collect more responses than expected, this would have been feasible. Reducing, and potentially eliminating, the limitations found in this experiment is future work.

[29] "According to you, what does this pictograph message mean?"

[30] "Do you think that your target user would be able to understand the meaning of this pictograph message?"

[31] "(Optional:) In case you indicated that your target user would not be able to understand the meaning of this pictograph message (well), why do you think that would be the case?"

In an attempt to reach a wide and diverse audience, the survey was hosted online. It was shared by a number of disability organisations on social media websites, such as Facebook and Twitter. We were able to collect 28 complete responses.

5.5.4.3.1 Characteristics of the Participants and Target Users A full description of the participants and their end users is given in Appendix E.1 and E.2.

Among the 28 participants who filled in the survey (see Figure 5.19), there are 10 teachers (35.71%), 9 carers (32.14%), 4 speech therapists (14.29%), 4 family members (14.29%), and 1 paramedic (3.57%). The average age of the target users is 22.8 years. At least[32] 16 target users (57.14%) go to a special needs school, whereas 6 users (21.43%) live in a residential group, 4 users (14.29%) go to a day centre, and 2 users (7.14%) attend a regular elementary school.

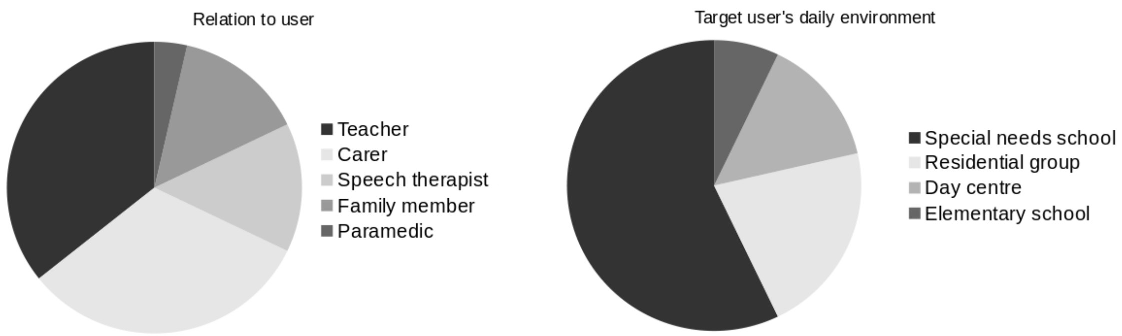


Figure 5.19: Profile of the participants and their target users.

We inquired the participants about the expressive and receptive skills of their target user (see Figure 5.20). 14 users (50%) use (simple) spoken language to express themselves, although writing difficulties may occur, while at least 12 users (42.86%) need pictographs to communicate, and 2 users (7.14%) experience severe expressive limitations. In their communication with the target user, 21 participants (75%) use spoken language without any form of additional gestural or pictograph support, although reading difficulties may occur, whereas 6 participants (21.43%) use (a combination of spoken language and) pictographs, especially to provide visual support for daily schedules and to explain social situations. 1 participant (3.57%) indicates that the user has severe receptive limitations. With the exception of 1 user (3.57%), all target users have prior experience with pictographs.

[32]Participants #13, #14, and #23 consider groups of users (of an unknown size).

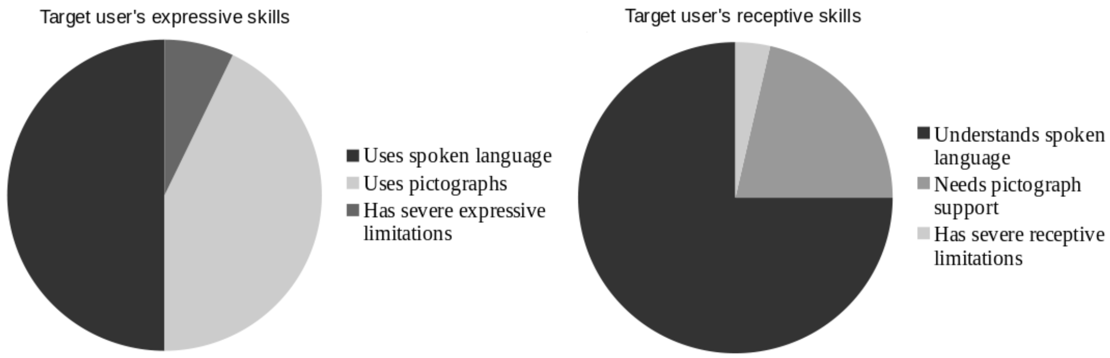


Figure 5.20: Target users' expressive and receptive skills.

With respect to technology use (see Figure 5.21), at least 16 target users (57.14%) have access to a computer, at least 15 users (53.57%) to a tablet, and at least 6 users (21.43%) to a smartphone, whereas at least 7 users (25%) use none of these technologies. Only 4 target users (14.29%) access social media websites on a daily basis, while at least 14 users (50%) never use social media websites.

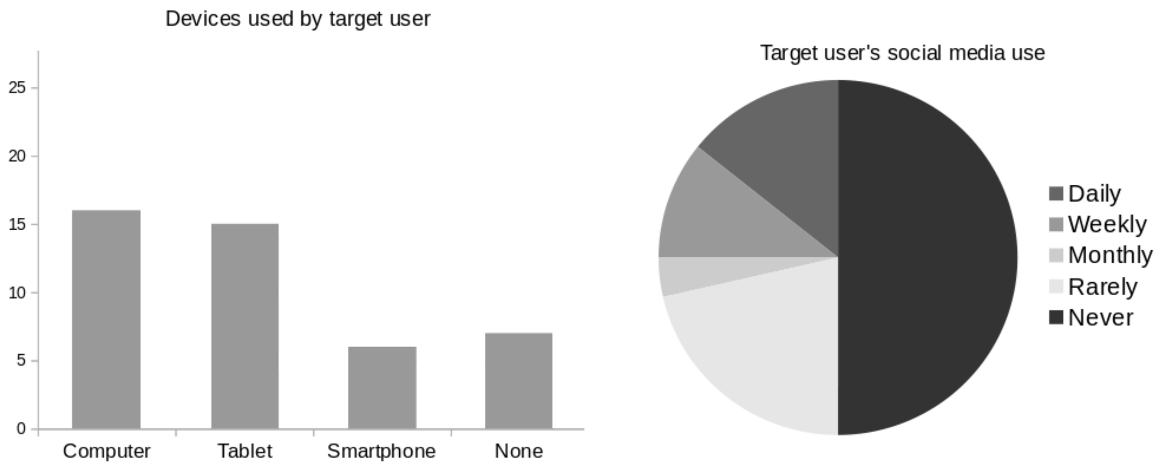


Figure 5.21: Use of technology by target users.

According to the participants, there are several reasons why social media websites are (or could be) beneficial for their target user, for instance, interacting with others (especially family and friends), watching and sharing pictures, watching videos and music clips, playing games, communicating about practicalities (such as a user's estimated time of arrival), and overall, boosting the user's self-esteem by being able to do the same things as his/her non-disabled peers.

Accessing social media websites, however, is difficult for all users, including those users who access the websites on a daily or weekly basis. 8 participants (28.57%) claim that social media websites are simply "too difficult" for their target user in general. Among the more specific reasons that are mentioned, 12 participants (42.86%) explicitly point out reading and writing issues, and the fact that the users' messages are often misunderstood by their non-disabled communication partners. 4 participants (14.29%) are worried about privacy issues, such as accepting friendship requests from strangers. Other obstacles are navigation issues, having no access to appropriate communication interfaces, and the abundant amount of information that may suddenly appear on the user's screen.

	Median	**# Correct trans.**	**# Incorrect trans.**
Passivity non-simp.	5.5	23	5
Passivity simp.	6	24	4
Relative non-simp.	5	16	12
Relative simp.	8	28	0
Subordination non-simp.	6	26	2
Subordination simp.	7	28	0
Coordination non-simp.	6	28	0
Coordination simp.	6	28	0
Appositive non-simp.	5	25	3
Appositive simp.	5	27	1
Passivity + subordination non-simp.	3	14	14
Passivity + subordination simp.	5	22	6

Table 5.9: Median of the difficulty ratings, and number of correct and incorrect translations for each condition.

5.5.4.3.2 Discussion of the Results We ask the participants to judge the difficulty of each pictograph sequence for their target user by picking a number on a 10-point ordinal scale, where 1 = *definitely too difficult for the user* and 10 = *definitely easy enough for my user* (and 5 = *rather difficult* and 6 = *rather easy*). We also ask the respondents to translate the pictograph sequence (see Table 5.9), and we provide them with the option to leave a comment with respect to the difficulty of the sequence.

For each syntactic phenomenon, we compare two pictograph sequences: a simplified variant and a non-simplified variant.

The Wilcoxon signed-rank test can be used for comparing two paired samples when the data are placed on an ordinal scale (McCrum-Gardner 2008). The hypothesis being tested is whether the median (see Table 5.9) difference is zero (as opposed to the mean difference in the paired *t*-test).

The Wilcoxon signed-rank test relies on the W-statistics.[33] For large samples with $n>10$ paired observations, the W-statistics approximate a normal distribution. This means that we can use the Z-value[34] to evaluate our hypotheses instead.[35] The p-value denotes significance level. The results per type of syntactic phenomenon are shown in Table 5.10.

	Z-value	**p-value level**
Passivity	-0.0933	-
Relative clause	-4	**
Subordination	-3.1492	**
Coordination	-0.4829	-
Appositive clause	-1.0601	-
Subordination + passivity	-2.8857	**

Table 5.10: Results of the Wilcoxon signed-rank test. $*p < 0.05, **p < 0.01$.

For each paired sample, we will discuss the results of the Wilcoxon signed-rank test, evaluate the amount of correct and incorrect translations per sample, and highlight a number of written remarks concerning the syntax or difficulty of the pictograph sequences. All remarks (in Dutch) and their translations are presented in Appendix E.3. We made a Beta and a Sclera version of the test (with identical pictograph sequences, i.e., the same amount of pictographs and the same lemmas), as some participants indicated a strong preference for one pictograph set over the other. For brevity, only Sclera examples are shown here.[36] Keep in mind that the lemmas (meanings) of the individual pictographs were also displayed in the survey.

[33]The W-statistic is used to assess agreement between different raters, and ranges from 0 (no agreement between all raters) to 1 (perfect agreement between all raters).

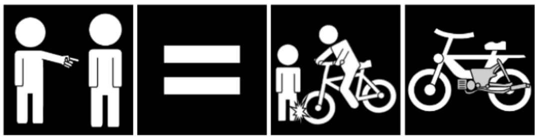
[34]The Z-test is used to determine whether two population means are different when the variances are known.

[35]http://www.statisticssolutions.com/how-to-conduct-the-wilcox-sign-test/

[36]All Beta sequences are available on request.

Passivity

Non-simplified



Je werd aangereden door een motor. 'You were hit by a motorcycle.'

Simplified

Ik word gepest door een meisje. 'I am being bullied by a girl.'

The non-simplified sequence: One participant, who provides an incorrect translation, comments on feeling unsure about whether the pictograph sequence is written in the active voice or in the passive voice. Similarly, two participants, who share an incorrect interpretation, and four participants, who present a correct translation, report that it is unclear who crashed into who.

The simplified sequence: One participant, who provides a correct translation, questions the identity of the bully. Similarly, one participant, who also presents a correct translation, asks himself/herself who the bully is in this scenario, claiming that, if the bully's identity is *I/me*, it would have been better if that particular pictograph came first in the sequence. This indicates that the participant believes that the agent (i.e., the bully) should be placed in front of the action (i.e., the verb *to bully*) to make the pictograph sequence easier to interpret. In this example, we show a simplified pictograph sequence in which the agent, as suggested, appears in front of the verb, as opposed to the agent in the non-simplified sequence.

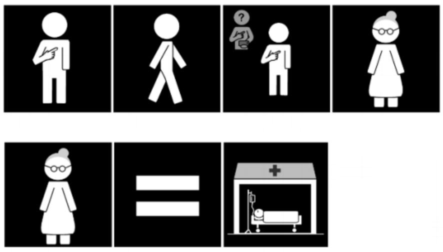
Discussion: While the difference between the difficulty scores that are assigned by the participants for the non-simplified and simplified sequence is not statistically significant according to the Wilcoxon signed-rank test, and while most of the participants manage to provide a correct translation for both pictograph sequences (23 versus 24 correct translations for the non-simplified and the simplified sequence, respectively), it is worth mentioning that 7 out of 28 participants in the non-simplified condition question the distribution of the agent and patient roles, whereas only two such comments are made in the simplified condition.

Relative clause

Non-simplified

De berichten die je me gestuurd hebt zijn goed. 'The messages that you sent me are good.'

Simplified



Ik ga naar mijn oma die in het ziekenhuis ligt. 'I am visiting my grandmother who is in the hospital.'

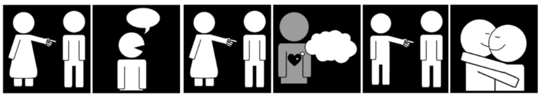
The non-simplified sequence: Four participants, who present an incorrect translation, and eight participants, who were able to get the translation right, comment that the pictograph sequence is too long for their user. Similarly, one participant, who gives an incorrect interpretation, comments that the order of the pictographs in the sequence confuses him/her. One participant, who provides an incomplete translation, reports that the syntax of the pictograph sequence is too complex, and proposes a different order: *You send me messages, the messages are good*.

The simplified sequence: Four participants, who provide a correct translation, confirm the hypothesis that short messages are easier to understand for their target user.

Discussion: Using the Wilcoxon signed-rank test, we find the difference between the difficulty scores that are assigned to the non-simplified and simplified sequence to be statistically highly significant. More evidence is given by the fact that 12 participants provide an incorrect translation for the non-simplified sequence, whereas no incorrect translations are given for the simplified sequence. The hypothesis that the non-simplified sequence is considered "too difficult" for most target users is also reflected in the comments.

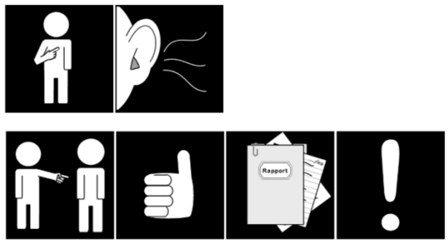
Subordination

Non-simplified



Hij zegt dat hij je een knuffel wil geven. ‘He says that he wants to give you a hug.’

Simplified



Ik hoorde dat je een goed rapport hebt! ‘I heard that you have good grades!’

The non-simplified sequence: One participant, who gives a correct interpretation, comments that the message in this sequence seems to be twofold, thus hinting that the message should be split into two parts. Five participants, who also provide a correct translation, remark that the message is too long or too complex for their target user.

The simplified sequence: Three participants, who provide a correct translation, agree that both messages are short, and therefore easier to understand.

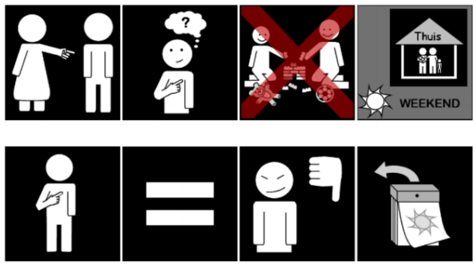
Discussion: Although most participants are able to provide a correct translation for both samples (with the exception of two erroneous translations in the non-simplified condition), the Wilcoxon signed-rank test indicates that the difference between the difficulty scores that are assigned to the non-simplified and simplified sequence is statistically highly significant. The complexity and difficulty of the non-simplified condition is also underlined in the comments.

Coordination

Non-simplified

Ik ben op woensdag een lieve baby gaan bezoeken en ik heb hem de papfles gegeven. ‘I went to visit a cute baby on Wednesday and I gave him the bottle.’

Simplified



Je mag niet komen spelen volgend weekend omdat ik stout geweest ben. ‘You can’t come over to play next weekend because I’ve been naughty.’

The non-simplified sequence: While all 28 participants manage to provide a correct translation, the “length” and “complexity” of the pictograph sequence are underlined by no less than 14 participants.

The simplified sequence: One participant, who provides a correct translation, remarks that he/she feels like there is a causal relationship between both sequences, while there is no pictograph indicating causality. Similarly, three participants, who were also able to get the translation right, admit that they did not completely understand the logical connection between the two pictograph sequences. The length of the individual sequences, however, is no longer considered an issue.

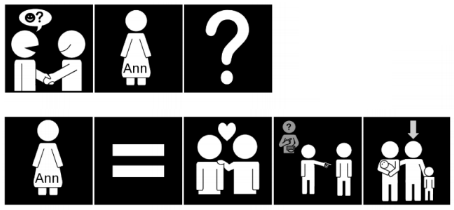
Discussion: The Wilcoxon signed-rank test does not indicate a significant difference between the difficulty scores that are assigned to the non-simplified and simplified sequence, and all translations that are provided by the participants are correct for both conditions. Based on the comments, however, we are able to make two observations. As expected, length is considered an issue in the non-simplified condition, but not necessarily in the simplified condition. However, by splitting the long pictograph sequence into two parts, there is a risk for the (logical) connection between both sequences to become less clear. Note that no pictographs are available for most function words, as they might be too abstract for some users.

Appositive clause

Non-simplified

We hebben Cars, de nieuwe film, gezien op televisie. 'We saw Cars, the new movie, on television.'

Simplified



Hoe is het met Ann, de vriendin van je papa? 'How's Ann, your father's girlfriend?'

The non-simplified sequence: Six participants, who provide a correct translation, and one participant, who shares an incorrect interpretation, report that the order of the pictographs in the sequence is confusing.

The simplified sequence: No comments are made with respect to the order of the pictographs or the length of the pictograph sequences.

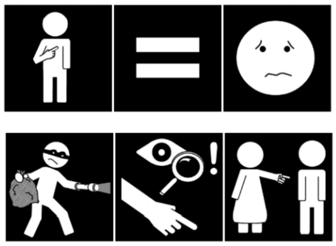
Discussion: Using the Wilcoxon signed-rank test, no significant difference is measured between the difficulty scores that are assigned to the non-simplified and the simplified sequence. Only three translations in the non-simplified variant and one translation in the simplified variant are incorrect. The comments reveal that at least 7 out of 28 participants believe that the order of the pictographs in the non-simplified condition is confusing, whereas no such comments are made in the simplified condition.

Subordination + passivity

Non-simplified

Ze zegt dat ze bedreigd werd door een jongen. 'She says that she was threatened by a boy.'

Simplified



Ik ben bang dat hij gevonden wordt door de dief. 'I'm scared that he will be discovered by the thief.'

The non-simplified sequence: Two participants, who present an erroneous translation, comment on feeling unsure about the distribution of the semantic roles and/or the order in which the pictographs are presented. The "length" and "complexity" of the sequence are highlighted by four participants, who provide a correct translation, and five participants, who give an incorrect translation.

The simplified sequence: Two participants, who provide a correct translation, remark that the order of the pictographs could be confusing for their target user.

Discussion: As an additional experiment, we introduce two sequences in which two syntactic phenomena to be simplified are presented: a subordinate clause and passive voice. Not only do we find half of the translations in the non-simplified condition to be erroneous (14 versus 22 correct translations), the Wilcoxon signed-rank test indicates that the difference between the difficulty scores that are assigned to the non-simplified and simplified sequence is statistically highly significant, as was also the case in the regular subordination experiment (but not in the passivity experiment). Most of the comments and erroneous translations in both conditions relate to the fact that the roles of the agent and patient seem unclear to the interpreter, although this issue is much more prominent in the non-simplified condition, with 10 out of 14 erroneous translations being the result of confusing the agent and patient roles.

5.5.4.3.3 Conclusion: User Studies Whereas the simplification of relative clauses, subordinate clauses, and clauses that present multiple syntactic phenomena to be simplified[37] were found to have statistically highly significant effects on the perceived difficulty of pictograph sentences, we were not able to find such results for simplified passive, coordinate, or appositive constructions. However, a qualitative analysis of the translation quality and the optional remarks has shown us that non-simplified passive, coordinate, and appositive constructions lead participants to provide a higher amount of incorrect translations, more semantic role confusion, and/or a larger amount of remarks, overall, with respect to pictograph order or syntactic complexity, even when these issues are not directly reflected in the assigned difficulty scores.

5.6 Advanced Simplification: Verb Group Simplification and Temporality Detection

Pictographs represent concepts without grammatical properties. As a result, temporal information gets lost during the translation process - unless the input sentence contains an explicit temporal modifier, such as *yesterday* or *next Monday*. Without the presence of such a modifier, the pictograph reader has no means of knowing whether the message is referring to the present, the future, or the past. For example, the pictograph translation shown in Figure 5.22, as generated by the baseline Text-to-Pictograph translation tool, may or may not refer to an event that takes place in the present. Still, the WAI-NOT corpus proves that our users like to talk about things that happened in their (recent) past, such as what they had for dinner, and their future plans, such as an activity that will take place the next day.

Temporality detection paves the way for verb group simplification.[38] The simplification module described in the previous sections clusters all verbs into a verb group, but it does not yet remove redundant pictographs, and it does not yet re-order the pictographs in a consistent way.

In section 5.6.1, we motivate our choice for depicting temporality by means of a separate pictograph, rather than augmenting the Sclera and Beta pictographs with grammatical markers. An overview of Dutch verb tenses is given in section 5.6.2. We present the verb group simplification and temporality detection modules in sec-

[37]In this case, we only evaluated subordinate clauses that contain a passive construction.

[38]The module described in this section was developed in collaboration with Tesselaar (2015).

tion 5.6.3. In our rule-based approach, the system attempts to match the input sentence's part-of-speech tags against an inventory of simplification rules. Finally, the results of the evaluation are shown in section 5.6.4.

Hij drinkt wijn / hij dronk wijn / hij zal wijn drinken.
He drinks wine / he drank wine / he shall wine drink.
'He drinks wine / he drank wine / he shall drink wine.'

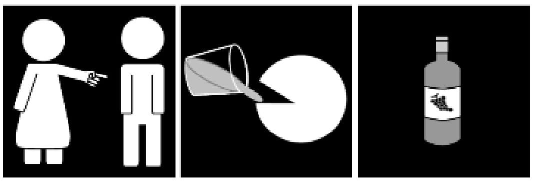


Figure 5.22: A pictograph translation without temporal indicators.

5.6.1 Depiction of Temporality in Pictographs

In Dutch, just like in many other languages, time is often described in terms of spatial dimensions and directions. This idea is reflected in the way in which we speak and think about time. For instance, we travel backward and forward in time, the future is ahead of us, and the past is behind us.[39] We may represent the past, the rewind option, or the previous slide by means of an arrow that points to the left, while the future, the fast-forward option, or the next slide can be depicted by means of an arrow that points to the right.

The pictographs in the SymWriter system[40] and Blissymbolics (Bliss 1965) are augmented with small symbols that correspond to a number of grammatical properties, such as number or the superlative degree. Tense is depicted by means of an arrow, which is shown on top of the verb's pictograph. In Blissymbolics, pictographs may also express passivity and conditionality. The advantage of this approach is that the length of the pictograph chain is not affected. The disadvantage is that the pictographs are more complex.

Yuizono et al. (2012) make use of separate tense pictographs that depict present, past, and future by means of an hourglass. In Visual Inter Lingua (VIL) (Leemans

[39]This is culturally dependent.
[40]http://www.widgit.com/

2001), tense is represented either implicitly in the meaning of the sentence when indicated by so-called *time-demonstratives* (*last*, *this*, *next*, etc.), or represented explicitly when indicated by so-called *time-pronouns* (*past*, *present*, *yesterday*, *tomorrow*, etc.). Time-pronouns are represented as a horizontal line with a small mark in the middle, which indicates the present, and a tiny arrow that is located in front of, on top of, or behind that mark, which indicates the past, present, or future.

In the pictograph translation tools, we make use of two existing sets, Beta and Sclera. Given that these sets were developed according to a number of design principles, we have chosen not to edit the original pictographs. Furthermore, the possibility of displaying tense information should be left optional, as it may be confusing for some groups of users.[41] For these reasons, we have decided to display tense as a separate pictograph. We do not create abstract representations, but we use the clock pictograph as a representation of time, since the users are already familiar with it. A small arrow is shown on top of the clock to represent past or future. If the action takes place in the present, no tense pictograph is shown. Conditionality is used to express a proposition whose validity is dependent on some condition, and may thus be very difficult to depict. For that reason, we represent it with a question mark. All three examples for Sclera are shown in Figure 5.23.

Figure 5.23: New Sclera pictographs for past, future, and conditionality.

5.6.2 Overview of Dutch Verb Tenses

According to the Algemene Nederlandse Spraakkunst (Haeseryn et al. 1997), Dutch has eight different tenses. The present tense and the past tense are simple forms. The other tenses are complex tenses, consisting of an auxiliary of time and a participle or

[41] In our user study on syntactic simplification (see section 5.5.4.3), 12 out of 28 participants indicated that displaying tense information, including conditionality, would be helpful for their target user. 7 out of 28 participants answered that their target user would be able to learn the meaning of the past and future markers, but might struggle with the conditionality marker. 9 out of 28 participants would not activate the temporal analysis module at all.

infinitive. Table 5.29 provides an overview. Temporality indicators and examples are shown in Table 5.30.

#	Tense	Dutch tags	Translation of tags
1	Simple present tense	WW(tgw)	V(pres)
2	Simple past tense	WW(verl)	V(past)
3	Present perfect tense	*hebben/zijn*(tgw) + VD/INF	*have/be*(pres) + past participle/INF
4	Past perfect tense	*hebben/zijn*(verl) + VD/INF	*have/be*(past) + past participle/INF
5	Simple future tense	*zullen*(tgw) + INF	*will*(pres) + INF
6	Future perfect tense	*zullen*(tgw) + VD + *hebben/zijn*(inf)	*will*(pres) + past participle + *have/be*(inf)
7	Simple past future tense	*zullen*(verl) + INF	*will*(past) + INF
8	Past future perfect tense	*zullen*(verl) + VD + *hebben/zijn*(inf)	*will*(past) + past participle + *have/be*(inf)

Table 5.29: An overview of the Dutch verb tenses.

Tense #	Temporality	Example	Translation
1	Present	*eet*	*eats*
2	Past	*at*	*ate*
3	Past	*heeft gegeten*	*has eaten*
4	Past	*had gegeten*	*had eaten*
5	Future	*zal eten*	*will eat*
6	Future	*zal gegeten hebben*	*will have eaten*
7	Conditional	*zou eten*	*would eat*
8	Conditional	*zou gegeten hebben*	*would have eaten*

Table 5.30: Temporality indicator and example for each of the eight Dutch verb tenses.

With respect to tense, Dutch distinguishes between present, past, future, and irrealis/conditionality. Every clause has its own temporal specification. For example, a main clause could be written in the present tense, while its subordinate clause or relative clause could refer to the past or the future. Therefore, the clause level is our point of departure for temporal analysis.

The aspect of a verb is determined by whether the action is completed or not. Since aspect is an abstract concept that is very difficult to depict, and since it does not add much information to the message, we have decided not to represent it in the translations.

5.6.3 Translating Dutch Verb Patterns into Pictographs

The temporal analysis module is located after the syntactic simplification module and its activation is entirely optional. The syntactically simplified output of the simplification module leaves the system with a number of independent clauses (i.e., simple sentences). Given that temporal analysis has to be performed at the clause level, these simple sentences are the logical starting point for the temporal analysis module. Note that, as a result of simplification, the verb pictographs have already been clustered in a verb group.

For every clause, the temporal analysis module first counts the number of verbs that are included in the verb group. Depending on that number, one out of three manually crafted rule sets is consulted (see Table 5.31).[42] The rules are checked in order. Each rule consists of a verb pattern on the left-hand side (condition) and a target structure on the right-hand side (result). An element on the left-hand side of a rule has the following structure: *(partial)tag_ID_lemma*. On the right-hand side, a tense indicator is shown for present (N, for *now*), past (P), future (F), and conditionality (C). Note that the N indicator will not be translated into a tense pictograph. Behind the tense indicator, a number of lemmas are shown in the *L_ID* format. The IDs on the right-hand side of the rules match the IDs on the left-hand side of the rules. In other words, the target side indicates the expected order of the pictographs, the tense pictograph to be generated, and the removal of pictographs. For instance, the auxiliaries *hebben* 'have', *zijn* 'be', *gaan* 'going to', and *zullen* 'will/shall' have no real semantic function, except for the encoding of temporal information. The tense pictographs cause these auxiliaries to become redundant. Modal verbs, on the other hand, are not removed, since there exist pictographs for most of them (especially in Beta), and since they contribute to the meaning of the message.

For example, rule (2-a) states that a finite form of the verbs *hebben* 'have' or *zijn* 'be' in the present tense, followed by any verb that is a past participle, leads to the generation of a past tense pictograph. The system preserves the past participle, but drops the auxiliary.

[42] A first version of these rules was developed by Tesselaar (2015). The table shows an updated version.

<table>
<tr><th>ID</th><th>Tense #</th><th>Condition</th><th>Result</th></tr>
<tr><td>(1-a)</td><td>1</td><td>WW(pv,tgw,_1_anyverb</td><td>N L_1</td></tr>
<tr><td>(1-b)</td><td>2</td><td>WW(pv,verl,_1_anyverb</td><td>P L_1</td></tr>
<tr><td>(1-c)</td><td>3/4</td><td>WW(vd_1_anyverb</td><td>P L_1</td></tr>
<tr><td>(2-a)</td><td>3</td><td>WW(pv,tgw,_1_hebben|zijn WW(vd,vrij_2</td><td>P L_2</td></tr>
<tr><td>(2-b)</td><td>4</td><td>WW(pv,verl,_1_hebben|zijn WW(vd,vrij_2</td><td>P L_2</td></tr>
<tr><td>(2-c)</td><td>1</td><td>WW(pv,tgw,_1_worden WW(vd,vrij_2</td><td>N L_2</td></tr>
<tr><td>(2-d)</td><td>2</td><td>WW(pv,verl,_1_worden WW(vd,vrij_2</td><td>P L_2</td></tr>
<tr><td>(2-e)</td><td>5</td><td>WW(pv,tgw,_1_zullen|gaan WW(inf,vrij_2</td><td>F L_2</td></tr>
<tr><td>(2-f)</td><td>1</td><td>WW(pv,tgw,_1_anyverb WW(inf,vrij_2</td><td>N L_1 L_2</td></tr>
<tr><td>(2-g)</td><td>7</td><td>WW(pv,verl,_1_zullen|gaan WW(inf,vrij_2</td><td>C L_2</td></tr>
<tr><td>(2-h)</td><td>2</td><td>WW(pv,verl,_1_anyverb WW(inf,vrij_2</td><td>P L_1 L_2</td></tr>
<tr><td>(3-a)</td><td>6</td><td>WW(pv,tgw_1_zullen|gaan WW(vd,vrij_2_anyverb WW(inf,vrij_3_zijn|hebben|worden|gaan| zullen</td><td>F L_2</td></tr>
<tr><td>(3-b)</td><td>8</td><td>WW(pv,verl_1_zullen|gaan WW(vd,vrij_2_anyverb WW(inf,vrij_3_zijn|hebben|worden|gaan|zullen</td><td>C L_2</td></tr>
<tr><td>(3-c)</td><td>3</td><td>WW(pv,tgw_1_anyverb WW(vd,vrij_2_anyverb WW(inf,vrij_3_zijn|hebben|worden|gaan| zullen</td><td>P L_1 L_2</td></tr>
<tr><td>(3-d)</td><td>4</td><td>WW(pv,verl_1_anyverb WW(vd,vrij_2_anyverb WW(inf,vrij_3_zijn|hebben|worden|gaan|zullen</td><td>P L_1 L_2</td></tr>
<tr><td>(3-e)</td><td>3</td><td>WW(pv,tgw_1_hebben|zijn WW(inf,vrij_2_blijven|komen|kunnen|moeten|mogen|willen WW(inf,vrij_3_anyverb</td><td>P L_2 L_3</td></tr>
<tr><td>(3-f)</td><td>4</td><td>WW(pv,verl_1_gaan|hebben|zijn WW(inf,vrij_2_blijven|komen|kunnen|moeten|mogen|willen WW(inf,vrij_3_anyverb</td><td>P L_2 L_3</td></tr>
<tr><td>(3-g)</td><td>5</td><td>WW(pv,tgw_1_zullen|gaan WW(inf,vrij_2_blijven|komen|kunnen|moeten|mogen|willen WW(inf,vrij_3_anyverb</td><td>F L_2 L_3</td></tr>
<tr><td>(3-h)</td><td>7</td><td>WW(pv,verl_1_zullen|gaan WW(inf,vrij_2_blijven|komen|kunnen|moeten|mogen|willen WW(inf,vrij_3_anyverb</td><td>C L_2 L_3</td></tr>
<tr><td>(3-i)</td><td>1</td><td>WW(pv,tgw_1_anyverb WW(inf,vrij_2_blijven|komen|kunnen|moeten|mogen|willen WW(inf,vrij_3_anyverb</td><td>N L_1 L_2 L_3</td></tr>
<tr><td>(3-j)</td><td>2</td><td>WW(pv,verl_1_anyverb WW(inf,vrij_2_blijven|komen|kunnen|moeten|mogen|willen WW(inf,vrij_3_anyverb</td><td>P L_1 L_2 L_3</td></tr>
<tr><td>(3-k)</td><td>1</td><td>WW(pv,tgw_1_anyverb WW(inf,vrij_2_anyverb WW(inf,vrij_3_zijn|hebben|worden|gaan|zullen</td><td>N L_1 L_2</td></tr>
<tr><td>(3-l)</td><td>2</td><td>WW(pv,verl_1_anyverb WW(inf,vrij_2_anyverb WW(inf,vrij_3_zijn|hebben|worden|gaan|zullen</td><td>P L_1 L_2</td></tr>
<tr><td>(3-m)</td><td>3</td><td>WW(pv,tgw_1_hebben|zijn WW(inf,vrij_2_anyverb WW(inf,vrij_3_zijn|hebben|worden|gaan|zullen</td><td>P L_2</td></tr>
<tr><td>(3-n)</td><td>5</td><td>WW(pv,tgw_1_zullen|gaan WW(inf,vrij_2_anyverb WW(inf,vrij_3_zijn|hebben|worden|gaan|zullen</td><td>F L_2</td></tr>
<tr><td>(3-o)</td><td>4</td><td>WW(pv,verl_1_hebben|zijn WW(inf,vrij_2_anyverb WW(inf,vrij_3_zijn|hebben|worden|gaan|zullen</td><td>P L_2</td></tr>
<tr><td>(3-p)</td><td>7</td><td>WW(pv,verl_1_zullen|gaan WW(inf,vrij_2_anyverb WW(inf,vrij_3_zijn|hebben|worden|gaan|zullen</td><td>P L_2</td></tr>
</table>

Table 5.31: The three complete rule sets for verb conversion and tense pictograph generation. The system uses the D-Coi tagset (Van Eynde 2005). Partial part-of-speech tags are used for matching.

Rule (3-g) states that a finite form of the verbs *zullen* 'will' or *gaan* 'go' in the present tense, followed by the infinitive form of any modal verb, followed by the infinitive form of any other verb, leads to the generation of a future tense pictograph. The auxiliary is dropped, while the modal verb and the other infinitive are preserved. An example of this is given in Figure 5.24.

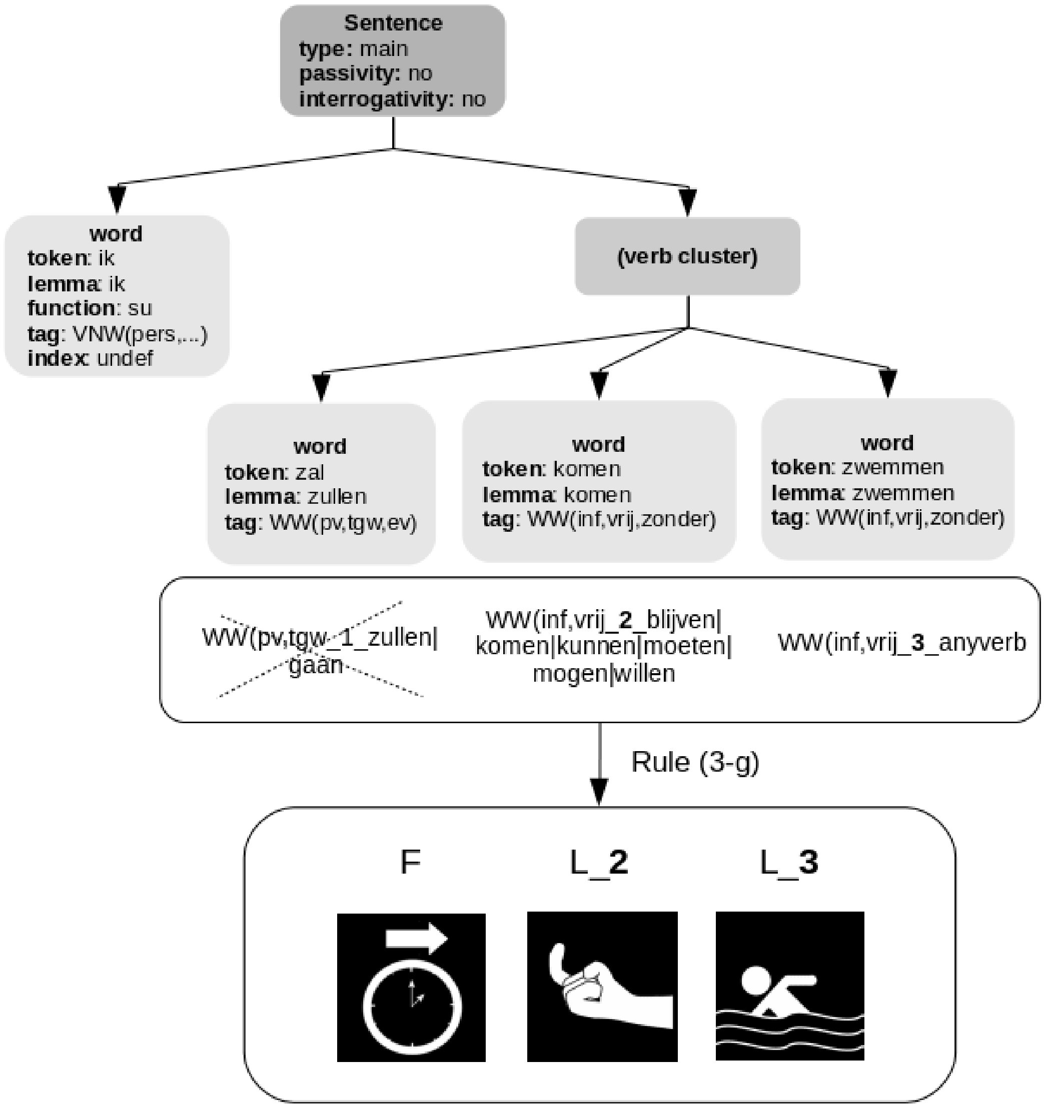


Figure 5.24: Verb group simplification and temporality detection of the sentence *Ik zal komen zwemmen* 'I'll come over to swim'.

The tense rules are not applied when more than three verbs are found. In principle, Dutch verb groups may contain an infinite amount of auxiliaries or modal verbs in the infinitive form. However, in practice, these complex constructions are rare.[43] As such, we do not create tense rules for them, and if we do encounter them, we leave the output of the syntactic simplification module as such.

Before converting the tense indicator into a tense pictograph, the system checks whether an explicit expression of time can be found. These expressions presented in Table 5.32.[44] If one of these adverbs or noun phrases is found, the tense pictograph is not generated, as it would be redundant and unnecessarily lengthen the sequence.

Expression of time	**Translation**	**Tense**
Vorig(e) week/maand/jaar	Last week/month/year	P
Eergisteren	The day before yesterday	P
Gisteren	Yesterday	P
Morgen	Tomorrow	F
Overmorgen	The day after tomorrow	F
Volgend(e) week/maand/jaar	Next week/month/year	F

Table 5.32: Expressions of time that prevent the generation of tense pictographs.

5.6.4 Evaluation

We use the 100 simplified sentences from De Standaard (see section 5.5.4.1) and apply the temporal analysis module to them (see Table 5.33). Note that any possible errors made by the syntactic simplification module are propagated, since temporal analysis is part of the simplification chain. The 100 simplified sentences correspond to a total of 265 clauses, for which temporal analysis is performed automatically. 257 clauses were assigned their appropriate temporal pictograph (future, past, conditional, or none for present tense). Verb simplification concerns possible changes in verb order and the removal of auxiliary verbs. 259 verb groups were simplified (or left intact, in 180 cases) correctly.[45] In two cases, an explicit temporal indicator was already present in the sentence, and no (redundant) temporal pictograph was

[43] Corpus research by Augustinus (2015) reveals that only 0.3% of all verb clusters (i.e., clusters consisting of two or more verbs) in Dutch contain four or more verbs.

[44] This list might not be exhaustive and can be extended.

[45] Without simplification, all verb groups would be left intact. In that case, the precision would be 67.92%.

generated. All errors can be attributed to erroneous syntactic parses, which lead to erroneous simplifications.

The WAI-NOT test set (see Appendix B.2) corresponds to a total of 213 independent clauses for which automated temporal analysis is performed.[46] Interestingly, no errors were made on this set; 213 verb groups were simplified (or left intact, in 183 cases) correctly (see Table 5.33).[47] This proves that our rule-based method for syntactic simplification and temporal analysis performs very well on the WAI-NOT email corpus.

	De Standaard		WAI-NOT	
Total amount of clauses	265		213	
Correct temporal pictograph generated (none for present)	257	(97.0%)	213	(100%)
Verb groups simplified (or left intact) correctly	259	(97.7%)	213	(100%)

Table 5.33: Evaluation of the temporal analysis module on 100 sentences from De Standaard and 50 messages from WAI-NOT.

Persona: Clemont (After)
By activating the syntactic simplification and compression modules, the Text-to-Pictograph translation engine has proven to be a particularly helpful tool for Clemont. It allows him to read his personal emails independently, without the help of a caregiver. Being able to get a grasp of his sister's life abroad makes Clemont feel more included, and Bonnie is excited that Clemont finally gets the privacy he desires. When Clemont becomes more familiar with the pictographs, his caregiver will disable the compression module, providing Clemont with richer - yet still simplified - pictograph translations. He is already looking forward to sending his sister his first pictograph-based reply.

5.7 Conclusion: Syntactic Simplification for Pictograph Translation

We described a syntactic simplification system for Text-to-Pictograph translation, the first of its kind. With no parallel data available, and given that parsers have become fast and reliable, we have opted for a rule-based approach that makes use of syntactic

[46]Note that these 213 clauses were not used during the development of the rule set.
[47]The precision without simplification would be 85.92%.

parsing. By using recursion and applying the simplification operations in a rational way, only one syntactic parse is needed per message. Promising results are obtained.

This experiment is a first step toward the creation of a text simplification tool for Dutch, which does not yet exist at this point. Note that, in order to develop such a system, lexical simplification should also be taken into account.

Chapter 6

Improvement #3: Word Sense Disambiguation for Pictograph Translation

The baseline Text-to-Pictograph translation system does not perform word sense disambiguation (WSD) to select the most appropriate sense of a word before converting it into a pictograph (see section 3.1). Instead, the most frequent sense of the word is chosen. This sometimes results in incorrect pictograph translations (see Figure 6.1).

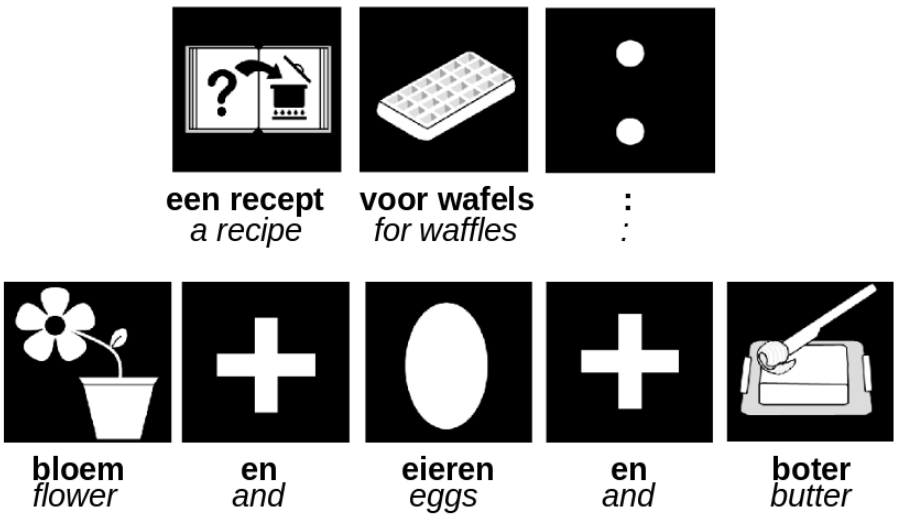


Figure 6.1: Example of erroneous Dutch-to-Sclera translation. The Dutch word *bloem* means both *flower* and *flour*. The most common sense is *flower*.

We describe the implementation of a WSD tool in the Dutch Text-to-Pictograph translation system.[1] After a discussion of related work (section 6.1), we present the semantic processing module of the Text-to-Pictograph translation engine and the WSD tool (section 6.2). We then describe the implementation of the WSD tool (section 6.3). Our evaluations show that improvements over the baseline in the Text-to-Pictograph translation tool are made (section 6.4). Finally, we conclude (section 6.5).

6.1 Status Quæstionis: Word Sense Disambiguation in Translation Applications

There are not many systems dedicated to the task of translating text for pictograph-supported communication. Mihalcea & Leong (2008) describe a system for the automatic construction of simple pictograph sentences. Their system uses a basic WSD tool that relies on WordNet as a lexical database. However, the system is focused on English and they do not evaluate the effectiveness of WSD within the context of a pictograph translation system.

SymWriter[2] allows users to insert arbitrary text, which is semi-automatically converted into pictographs. However, it does not provide automatic translation aids based on linguistic knowledge to properly disambiguate lexical ambiguities, which can lead to erroneous translations (Vandeghinste 2012).

There is contradictory evidence that natural language processing tools and information retrieval tasks benefit from WSD. Within the field of machine translation, early attempts to integrate WSD components met with limited success. Carpuat & Wu (2005) argue that it is difficult to use WSD models to obtain significant improvements for statistical machine translation tasks, even when supervised WSD models are used. Vickrey et al. (2005) and Neale et al. (2016), on the other hand, show that proper incorporation of WSD can lead to an increase in translation performance for automatic translation systems. Navigli (2009) underlines the general agreement that WSD needs to show its relevance *in vivo*: Full-fledged applications must be built including WSD either as an integrated or a pluggable component.

In this chapter, we implement WSD and evaluate its added value within the Text-to-Pictograph translation pipeline.

[1] Previous versions of sections 6.1 to 6.5 appeared in Sevens et al. (2016a).

[2] http://www.widgit.com/products/symwriter/

6.2 Description of the Tools

The following sections describe the architecture of the semantic module of the Text-to-Pictograph translation system (section 6.2.1) and the WSD tool (section 6.2.2).

6.2.1 Semantic Processing in the Baseline Text-to-Pictograph Translation System

For each word in the source text, the baseline Text-to-Pictograph translation system returns all WordNet synset identifiers that are connected to that word. The synsets are filtered, keeping only those where the part-of-speech tag of the synset matches the part-of-speech tag of the word. This way, the system is able to deal effectively with the semantic ambiguity of words across different parts-of-speech (such as the noun *kom* 'bowl' and the verb *kom* 'come').

The WordNet synsets are used to connect pictographs to natural language text. The system might be able to retrieve multiple pictographs for a given input word. There is a chance that the baseline system is confronted with an equally likely choice between two or more pictographs, which correspond to different meanings of the same word (see Figure 6.2). In that case, the most commonly occurring sense according to DutchSemCor (Vossen et al. 2012), a one-million word sense-tagged Dutch corpus, is chosen.

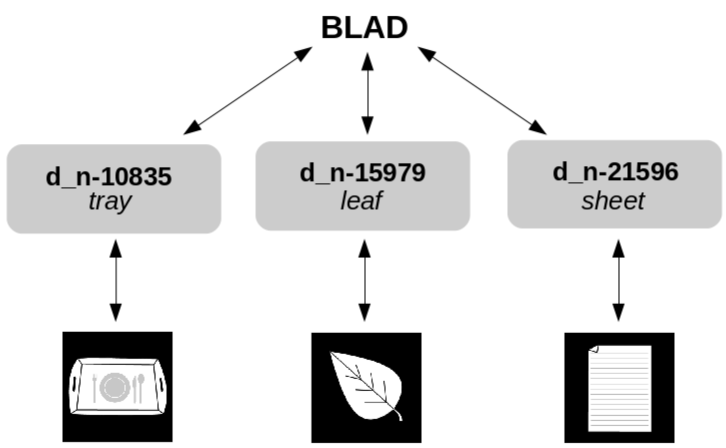


Figure 6.2: The Dutch word *blad* is linked to three different pictographs through its synsets.

6.2.2 The Word Sense Disambiguation Tool

We implement the Dutch WSD tool that was developed by Rubén Izquiero[3] within the framework of the DutchSemCor project (Vossen et al. 2012).

DutchSemCor delivers a one-million word Dutch corpus that is fully sense-tagged with senses and domain names from the Cornetto database. It is constructed as a balanced-sense lexical sample for the 3,000 most frequent and polysemous Dutch words, with about 100 examples for each sense. Part of the corpus was built semi-automatically and other parts manually.

In the first phase of the project, 282,503 tokens for 2,870 nouns, verbs and adjectives - corresponding to a total of 11,982 senses - were annotated by two annotators. For each sense, they manually selected 25 diverse examples. The remainder of the corpus was automatically tagged by a supervised WSD system, which was built using the manually tagged data. The supervised system looked for the remaining 75 representative examples for each sense to complete the corpus. Low-confidence examples were manually validated by the annotators. In the last phase, even more examples were added to represent the contextual variety and sense distribution of the polysemous words as reflected in external corpora.

The resulting WSD system was built from this sense-annotated corpus. It is based on *k*-nearest neighbour classification, for which the authors determined an optimal set of features (word, lemma, part-of-speech, and bag-of-words). Different sizes for the context were considered, ranging from one to five context words. The feature set that led to the best performance (81.62% token accuracy) is the one that uses words that appear in a 1-token window around the target word, in combination with a bag-of-words representation of these context words. In other words, the sense-discriminating context words either appear directly in front of the target word to be classified, or directly behind it.

This WSD system takes natural language text as input and returns the confidence values of all senses according to support vector machines.[4] Note that senses correspond to Cornetto synsets in both the Text-to-Pictograph translation tool and the WSD system.

[3]https://github.com/cltl/svm_wsd

[4]For a more detailed explanation on how this WSD system was built and tuned, we refer to Vossen et al. (2012).

6.3 Implementation

The WSD tool is activated after the linguistic analysis and synset retrieval steps. Instead of outputting only one winning sense per word, we adapted the WSD tool to output the scores of every possible sense of the target word. As mentioned above, in the Text-to-Pictograph translation system, senses correspond to synsets which are attached to the input words. The WSD scores are added as features of these synsets.

Next, we adapt the A* path-finding algorithm to include the WSD score in the penalty calculation as a bonus: A high WSD score biases the selection of the pictograph toward the winning sense. The score is weighted by a trainable parameter, which we call the *WSD weight*, to determine the importance of WSD in relation to the other parameters. The WSD score of each sense, which is multiplied by this parameter, functions as a bonus: It is subtracted from the initial penalty (which has a standard value of 0) that is attributed to each synset (see Equation 6.1).

$$Initialpenalty = \begin{cases} 0 - (wsdweight \times wsdscore), & \text{if WSD score} \\ 0, & \text{otherwise} \end{cases} \tag{6.1}$$

In other words, if the baseline system is confronted with an equally likely choice between two or more pictographs, the synset with the highest WSD score will cause the initial (and therefore, overall) penalty to be lower, and that synset (and its attached pictograph) will be favoured over other synsets (and their attached pictographs) by the A* algorithm.

We have tuned the parameters through an automated procedure. The original tuning corpus consists of 50 messages from the WAI-NOT corpus (see Appendix B.1), which we manually translated into Beta and Sclera pictographs. To the original tuning corpus, we added five more hand-picked messages from the corpus that include a polysemous word (that has pictographs linked to at least two of its synsets). Biasing the tuning corpus like this was necessary, since the original tuning set has very few ambiguous words.[5]

We tune the parameters using a local hill climber (Vandeghinste et al. 2017), following the procedure described in section 3.2.2. The new parameter values are presented in Table 6.1.

[5] Only two examples were found.

Parameter	Min	Max	Step	Sclera	Beta
Cornetto relations					
Threshold	5	20	1	11	8
Hyperonym penalty	0	15	1	4	7
XPos penalty	0	15	1	3	6
Antonym penalty	0	15	1	2	7
Pictograph features					
Wrong number	0	10	1	4	2
No number	0	10	1	6	9
Word Sense Disambiguation					
WSD weight	0	10	1	2	2

Table 6.1: Parameter values for the Text-to-Pictograph translation system with WSD implementation after tuning.

6.4 Evaluation

For the task of measuring the impact of the WSD tool on pictograph translation quality, our test set (see Appendix B.2) is too small. It does not contain many polysemous words with multiple senses that are linked to pictographs: We were able to find only six polysemous words in the test set.

Alternatively, we will evaluate the effects of the WSD tool by means of the **test point method** (Shiwen 1993). A test point is a specific problem which an MT system has to resolve. In the test point method, for each test sentence, substring matching is used to determine if the specific test point has been correctly processed.

We selected 50 sentences from the WAI-NOT corpus that contain a word that has at least two pictographs attached to its synsets (belonging to the same grammatical category) and manually calculated the precision of their pictograph translations, while focussing on the ambiguous words, before and after implementing the WSD tool. The polysemous words that we searched for in the corpus are shown in Table 6.2. An example of a sentence containing two polysemous words is given in Example 6.1 and Figure 6.3.[6]

For Beta, choosing the most frequent sense for each word gives us a correct translation for 28 out of 50 ambiguous words, while the addition of the WSD tool leads to a correct translation for 42 out of 50 words. For Sclera, we get 29 out of 50 correct

[6]Note that we do not apply syntactic simplification in this example.

Word	Sense	Word	Sense
Bal	Sense 1: ball (toy)	**Kleed**	Sense 1: dress
	Sense 2: ball (dance party)		Sense 2: carpet
Bank	Sense 1: bank	**Knuffel**	Sense 1: hug
	Sense 2: brench		Sense 2: cuddly toy
Been	Sense 1: leg	**Licht**	Sense 1: light (not heavy)
	Sense 2: bone		Sense 2: light (not dark)
Blad	Sense 1: leaf	**Muis**	Sense 1: mouse (animal)
	Sense 2: tray		Sense 2: mouse (computer)
	Sense 3: sheet	**Nagel**	Sense 1: nail (finger)
Blik	Sense 1: tin/can		Sense 2: nail (fastener)
	Sense 2: look	**Noot**	Sense 1: nut
Bloem	Sense 1: flower		Sense 2: note (music)
	Sense 2: flour	**Pad**	Sense 1: toad
Bus	Sense 1: bus		Sense 2: path
	Sense 2: mailbox	**Schat**	Sense 1: darling
Das	Sense 1: badger		Sense 2: treasure
	Sense 2: tie	**Tas**	Sense 1: cup
Golf	Sense 1: golf		Sense 2: bag
	Sense 2: wave	**Tong**	Sense 1: tongue
IJs	Sense 1: ice cream		Sense 2: sole (fish)
	Sense 2: ice		

Table 6.2: The polysemous words that were targeted during the evaluations. Each sense is linked to a different pictograph in the Cornetto database.

translations for the most frequent sense condition, and 41 out of 50 correct translations for the WSD condition. We manually checked the results and found that all errors can be attributed to the WSD tool, which sometimes favours the wrong sense. Note that these results approximate the 81.62% token accuracy that is reported by Vossen et al. (2012).

(6.1) (...) dat in de herfst de **blaadjes** vallen en de eekhoorn **nootjes** zoekt.
(...) that in the autumn the leaves fall and the squirrel nuts seeks.
'(...) that, in the autumn, the leaves fall down and the squirrel looks for nuts.'

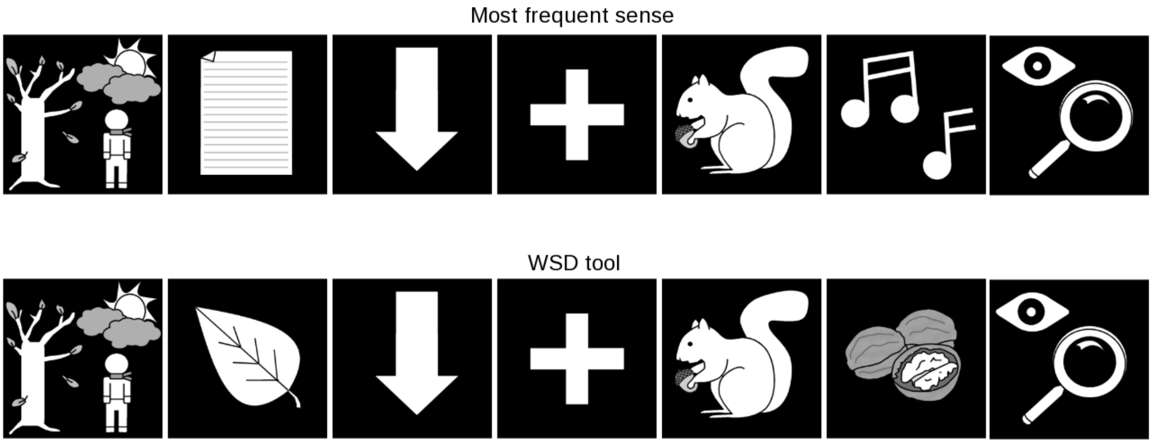


Figure 6.3: Translations of Example 6.1, using the most frequent sense (top) and the WSD tool (bottom).

6.5 Conclusion: Word Sense Disambiguation for Pictograph Translation

We implemented and evaluated the effect of WSD on the Text-to-Pictograph translation system for Dutch. Improvements over the baseline system were made. We can affirm that disambiguation works in most cases where multiple senses of ambiguous words are linked to pictographs in the lexical database. The system with WSD is now less likely to pick the wrong pictograph for an ambiguous word, effectively improving pictograph communication for the target users.

CHAPTER 7

Evaluating the Improved Text-to-Pictograph Pipeline

In the previous chapters, we described a Text-to-Pictograph translation tool for Dutch Internet users with ID. In the improved Text-to-Pictograph pipeline, input text first undergoes advanced, context-sensitive spelling correction (see Chapter 4). By introducing deep linguistic analysis in the translation process, we paved the way for syntactic simplification of complex input text (see Chapter 5). The advanced verb simplification module and temporality detection tool remove redundant verbs from the pictograph chain and generate tense pictographs in the output sequence. Finally, the baseline semantic analysis step is enhanced with a word sense disambiguation module (see Chapter 6) that biases the selection of the pictograph toward the winning sense. Each of these three modules (spelling correction, syntactic simplification, and word sense disambiguation) can be (de)activated by means of a switch.

In this chapter, we evaluate the improved Text-to-Pictograph translation pipeline. Evaluations using automated metrics are difficult, primarily because syntactic simplification was not part of the baseline translation system, in the first place. Using manually simplified reference translations to estimate translation quality would unfairly disadvantage the original system. Therefore, we opt for human evaluations, instead. Section 7.1 presents the set-up of the user survey and describes the participants. An analysis of the results is shown in section 7.2. We perform two types of evaluations: an objective one, in which we measure the accuracy of the transcriptions, and a subjective one, in which we estimate the quality of the transcriptions. Section 7.3 concludes.

7.1 Description of the Survey

We took 200 random emails from the WAI-NOT corpus (see section 2.2.4.2) and we translated them into Sclera[1] pictographs using two different versions of the Text-to-Pictograph translation engine. The first version is the baseline system (see Chapter 3), which uses the context-insensitive spelling corrector. This system does not apply syntactic simplification or temporality detection, and it does not perform word sense disambiguation. The second version is the improved Text-to-Pictograph translation pipeline, which includes the automated spelling corrector for people with ID, the syntactic simplification and temporality detection modules, and the word sense disambiguation tool. After translating the complete set, we found that only 21 out of 200 translations were identical in both versions of the translation engine.

Sentence #	Original WAI-NOT messages
#1	ya de school is leuk ik mis ye ook en ya ik ga praaten met ye
#2	juffrouw *Name* ik kom naar de school fiuf op 15 maart
#3	ye weet dat *Name* dik is en zij is niet mooi yij bent mooier.
#4	Ik ben blij dat morgen school is want vakantie vas niet leuk met mijn broer
#5	ik haa morgn in de sneeuw speelun
#6	nee *Name* gij bent een aap *Name*
#7	kom jij morgen en alles gaat goed met mij en ik niet want ik ben ziek en ik zit thuis
#8	ja ik heb nog altijt konijn en ik ga ook in maart krijgen een hond van mijn mama
#9	juf *Name* ik mis je maar op maandag is school en op vrijdag was niet leuk door *Name* en *Name*
#10	ik mag als doen wat ik wul
#11	ik vind je een leuk vriendin kom je volgende week naar school ja of nee
#12	ga je me belen want jij heb da ge zegd
#13	ik ben 12jaar en op juni ben ik jaareg en dan woort ik 13jaar maar nu ben ik nog 12jaar
#14	ik ben ziek dus ga ik slaapen
#15	ja oke en ik ben beroemt

Table 7.1: Test set for the evaluation of the improved Text-to-Pictograph translation pipeline.

From the 179 remaining cases, we randomly selected 15 sentences with a minimal length of 5 words (see Table 7.1). We created two surveys: one containing either a baseline pictograph translation or a pipeline pictograph translation for each of the 15 sentences,[2] and another one containing the remaining 15 translations.[3] For words

[1]We asked eight translators who are (to some extent) familiar with the Sclera set to participate in the experiment, hence our choice for Sclera.

[2]Version A of the survey, which can be found here: https://goo.gl/forms/Cafc1vyTUker8BOn1

[3]Version B of the survey, which can be found here: https://goo.gl/forms/kmYok3WaBpQzy9dA3

that could not be translated into a pictograph (for instance, due to the spelling corrector not being able to provide a correct form), we showed a black pictograph. This represents the idea that the untranslated word is "unreadable" for a person with reading difficulties. We did not show the lemmas (meanings) of the pictographs, as the lemma could provide a hint as to where the translation engine went wrong.

We divided a group of eight participants - scholars who are (to some extent) familiar with the Sclera set - into two groups: Four participants filled in one version of the survey (version A), while the other four participants filled in the other version of the survey (version B).[4] In dividing the participants into two groups, we made sure that an equal amount of "expert" Sclera users (i.e., scholars who helped establishing or improving the pictograph links) and "competent" Sclera users (i.e., scholars who read papers about the project and attended several lectures on the topic of Text-to-Pictograph translation) were included in each group, based on their previous direct involvement in the pictograph translation project (see Table 7.2). We then asked the participants to provide a transcription for each of the 15 pictograph sequences.

	Survey version A (8 baseline translations) (7 pipeline translations)	**Survey version B** (7 baseline translations) (8 pipeline translations)
Expert Sclera users	2	2
Competent Sclera users	2	2

Table 7.2: Distribution of "expert" and "competent" participants across the two versions of the survey.

7.2 Evaluation

We perform two types of evaluations. The first type of evaluation is an objective evaluation, in which we measure the overlap between the content words in the WAI-NOT messages and the content words in the transcriptions provided by the participants (section 7.2.1). The second type of evaluation is a subjective evaluation with one judge (i.e., the author) of the overall quality of the transcriptions (section 7.2.2).

[4] In other words, for each WAI-NOT sentence, four participants were shown a baseline translation, and the other four participants were shown a pipeline pictograph translation.

7.2.1 Objective Evaluation of the Accuracy of the Transcriptions

We calculate the accuracy of the transcriptions that are provided by the participants by measuring the overlap between the content words in the transcriptions of the participants and the content words in the original input messages; for each content word in the input message, we manually verify whether it was also included in the participant's transcription. Synonyms of the content words are considered correct translations, as well. Note that, in this type of evaluation, we do not take the combined semantics (i.e., the meaning of the whole sentence) into account. Table 7.3 and Figure 7.1 show the overall accuracy of the transcriptions per type of pictograph translation (i.e., baseline versus pipeline) for each participant. The overall accuracy is obtained by dividing the total amount of content words that were correctly included in the participant's transcriptions by the total amount of content words that *should* have been included in those transcriptions. We calculate significance between the baseline condition and the pipeline condition per participant using the paired samples t-test. This test can be used to compare two paired samples when data are placed on an interval scale (McCrum-Gardner 2008). The result is statistically highly significant: The value of t is 7.04,[5] and the value of p is 0.0002. In other words, participants consistently provided more **accurate** transcriptions for the pipeline pictograph translations than for the baseline pictograph translations.

We also calculate the overall accuracy of the translations per type of pictograph translation (i.e., baseline versus pipeline) for each sentence. The overall accuracy is obtained by dividing the total amount of content words that were correctly included in the participants' transcriptions for that sentence by the total amount of content words that *should* have been included in those transcriptions. The results are presented in Table 7.3 and Figure 7.1. Using the paired samples t-test, we find the difference between the transcriptions provided for the baseline pictograph translations and the pipeline pictograph translations per sentence to be statistically highly significant: The value of t is 3.06 and the value of p is 0.008. In other words, across all sentences, more **accurate** transcriptions were given for the pipeline pictograph translations than for the baseline pictograph translations.

[5] The t-value measures the size of the difference relative to the variation in the sample data.

Participant #	Mean acc. of baseline	Mean acc. of pipeline	Sentence #	Mean acc. of baseline	Mean acc. of pipeline
1	79.3%	84.8%	1	62.5%	83.3%
2	81.7%	89.4%	2	81.3%	100.0%
3	41.7%	68.1%	3	79.2%	87.5%
4	69.0%	83.1%	4	75.0%	77.1%
5	82.1%	96.7%	5	75.0%	93.8%
6	74.2%	83.6%	6	100.0%	100.0%
7	80.1%	97.0%	7	90.4%	84.6%
8	67.7%	85.8%	8	80.0%	97.5%
			9	73.2%	94.6%
			10	37.5%	20.8%
			11	53.9%	92.3%
			12	42.9%	67.9%
			13	33.8%	100.0%
			14	100.0%	100.0%
			15	80.0%	100.0%
Total	72.0%	86.1%	**Total**	71.0%	86.6%

Table 7.3: Overall accuracy of the transcriptions per type of pictograph translation (baseline or pipeline) for each participant and for each sentence.

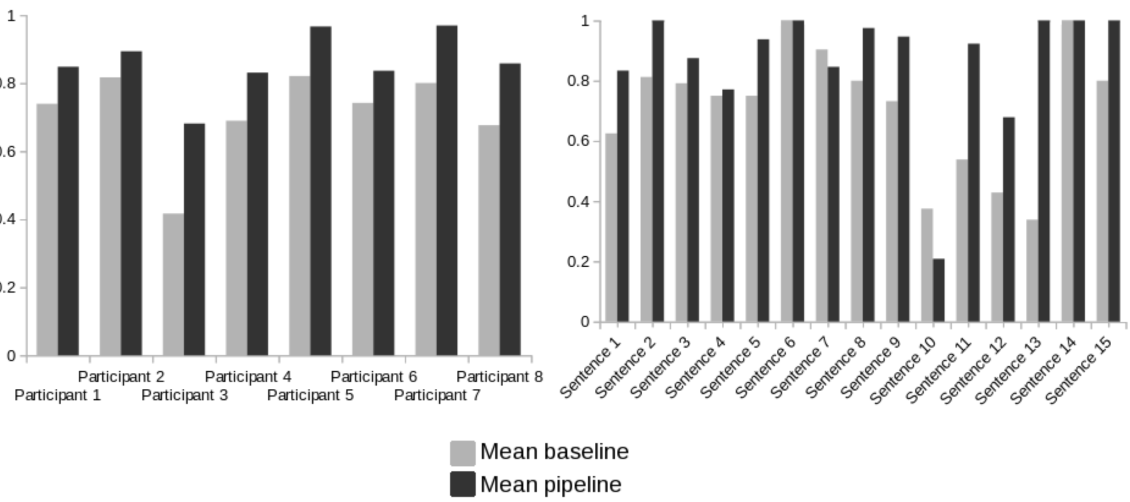


Figure 7.1: Graphical representations of the results from Table 7.3.

7.2.2 Subjective Evaluation of Transcription Quality

The second evaluation is a subjective, blind evaluation with one judge (i.e., the author) of the overall quality of the transcriptions, i.e. the combined semantics, using adequacy as a measure to indicate how much of the meaning expressed in the original WAI-NOT messages is also expressed in the target transcriptions. It is measured by means of a 5-point scale:

- 5: All meaning is preserved
- 4: Most meaning is preserved
- 3: Some meaning is preserved
- 2: Little meaning is preserved
- 1: No meaning is preserved

We first show the overall adequacy of the transcriptions per type of pictograph translation (i.e., baseline versus pipeline) for each participant. The overall adequacy is obtained by taking the median of the adequacy ratings that were attributed to the transcriptions made by each participant, for either baseline or pipeline translations. The results are shown in Table 7.4 and Figure 7.2. We calculate significance between the baseline condition and the pipeline condition per participant using the Wilcoxon signed-rank test. This test can be used to compare two paired samples when data are placed on an ordinal scale (McCrum-Gardner 2008) (see section 5.5.4.3.2 for a more detailed description of this test). The Wilcoxon signed-rank test relies on the W-statistics samples with n<10 paired observations. In this case, the W-value is 0. The critical value of W at n=8 and p<=0.01 is 3. Therefore, this result is highly significant. In other words, participants constantly provided more **adequate** transcriptions for the pipeline pictograph translations than for the baseline pictograph translations.

Table 7.4 and Figure 7.2 also show the overall adequacy of the translations per type of pictograph translation (i.e., baseline versus pipeline) for each sentence. The overall adequacy is obtained by taking the median of the adequacy ratings that are attributed to the transcriptions for each sentence, for either baseline or pipeline translations. For samples with n>10 paired observations, the W-statistics approximate a normal distribution. This implies that we can use the Z-value, instead. Again, the result is statistically highly significant: the Z-value is -2.93, whereas the p-value is 0.003. In other words, across all sentences, more **adequate** transcriptions were given for the pipeline pictograph translations than for the baseline pictograph translations.

Participant #	Median adeq. of baseline	Median adeq. of pipeline	Sentence #	Median adeq. of baseline	Median adeq. of pipeline
1	3	5	1	3	4.5
2	2.5	5	2	3.5	5
3	2	5	3	2	4
4	2.5	4	4	3	3.5
5	3	5	5	2	5
6	3	4	6	5	5
7	3	4.5	7	3	4
8	2	4	8	3	5
			9	4	4
			10	1	1
			11	2.5	4
			12	1	4.5
			13	1	5
			14	5	5
			15	3	5
Total	2.75	4.5	**Total**	3	4.5

Table 7.4: Overall adequacy of the transcriptions per type of pictograph translation (baseline or pipeline) for each participant and for each sentence.

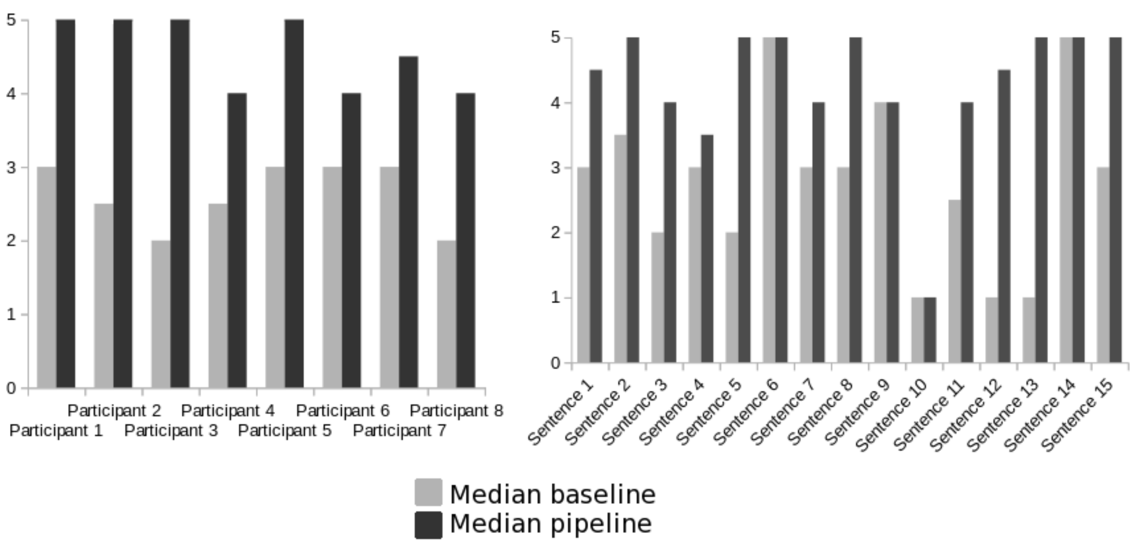


Figure 7.2: Graphical representations of the results from Table 7.4.

7.3 Conclusion: Evaluating the Improved Pipeline

Using automated evaluations, manual evaluations, and user evaluations (where possible), we revealed a number of shortcomings of the baseline Dutch Text-to-Pictograph translation engine. Taking personal customisability into account, we developed three modules (spelling correction, syntactic simplification, and word sense disambiguation) that can be (de)activated by means of a switch. Each of these modules was evaluated separately, and promising results were obtained. In this chapter, as a final experiment, we evaluated the complete Text-to-Pictograph translation pipeline, using human transcriptions as an indicator of relative improvement. Our user tests showed that more accurate and more adequate translations can be obtained by activating the newly developed modules, thus effectively facilitating the comprehension of written text for people with reading difficulties in online environments.

Persona: Link (After)
With the help of his parents, Link has activated the spelling correction, syntactic simplification, and word sense disambiguation modules. He notices a huge improvement when using the Text-to-Pictograph translation tool on Facebook, especially when the input is long and complex, or when translating user-generated content. The temporal analysis module has proven to be particularly useful when Link is chatting with his friends and family, as the tool helps him understand whether their stories have already taken place, or whether they are talking about their plans for the coming days. Link feels more included and confident, knowing that captions, comments, and private messages are henceforth easier for him to understand.

Part III

Pictograph-to-Text Translation

Outline

The Pictograph-to-Text translation system provides help in *constructing* Dutch written messages by allowing the user to input (a combination of written text and) pictographs, and translates these messages into natural language text. It can be seen as the inverse translation engine of the Text-to-Pictograph translation tool, although not the same techniques can be applied.

The main challenges for the Pictograph-to-Text translation stem from precisely those features that make pictograph communication so attractive in the first place (Lemmens 2016:5). Pictographs are underspecified, both semantically and grammatically. That is, they individually encode less information than do corresponding words in natural languages. In the second place, pictographs can be used in any order, with the result that the input to translation may be unpredictable and potentially ambiguous. Furthermore, since there exist no corpora of pictograph language, traditional data-driven translation methods are difficult to apply.

In Chapter 8, we describe the pictograph input interface, which consists of two parts: a static hierarchy of pictographs, i.e., a three-level category system that contains several thousands of pictographs for the user to choose from, and a dynamic prediction system, which suggests contextually relevant or semantically related pictographs to the user. We then present the actual Pictograph-to-Text translation tool in Chapter 9. We discuss a variety of approaches, including language modelling and machine translation techniques, toward the generation of rich natural language text from underspecified pictograph input, while highlighting the advantages and disadvantages of each approach.

Chapter 8

Design of a User-Oriented Pictograph Input Interface

With the Pictograph-to-Text translation engine (see Chapter 9) relying on (a combination of written text and) pictograph input, the target user should be able to find and select his/her desired pictographs with ease. We have developed two different input methods. The first method is a static hierarchy of pictographs, which contains thousands of pictographs for the user to choose from (see section 8.1), whereas the second method suggests contextually relevant pictographs to the user, based on the pictographs that have already been inserted (see section 8.2). Note that these two input methods are not mutually exclusive; they are designed to be used in combination with each other.

Persona: Vivi (Before)
Freya's daughter Vivi has Down Syndrome. As a result of her disability, she often finds it difficult to verbally express her thoughts and ideas. Vivi currently works at a sheltered workshop, just a few blocks away from home. Over the past few months, she has taken a shuttle bus to and from work, but since the weather has been very nice lately, Vivi would like to travel on foot. Freya is feeling a bit worried, because phone calls are difficult for Vivi - especially when there is a lot of traffic noise. She is looking for new and accessible ways to facilitate communication with her daughter at distance.

8.1 A Static Hierarchy of Pictographs

During the development of the static pictograph hierarchy, the target users' involvement is twofold.[1] On the one hand, a large corpus of user-generated content, the WAI-NOT corpus (see section 2.2.4.2), is used to infer linguistic knowledge and to create a hierarchy that is tailored toward the end users' vocabulary and interests. On the other hand, during hands-on sessions with the end users, limitations and expectations are discussed and the new hierarchy is put to the test.

Section 8.1.1 gives an overview on the topic of pictograph categorisation systems. Our interface is developed according to the principles of user-centered design, which we describe in section 8.1.2. We discuss the shortcomings of the original WAI-NOT pictograph hierarchy, and we propose a method for building a more appropriate hierarchy in section 8.1.3. Section 8.1.4 presents the experiences and opinions of the end users. Finally, we conclude in section 8.1.5.

8.1.1 Status Quæstionis: Categorisation of Concepts

Literature on the topic of pictograph categorisation is sparse and the hierarchical structure of pictograph interfaces is often not well motivated. Yuizono et al. (2012), for instance, classify their 335 pictographs into grammatical categories, such as *adverbs*, *tense*, and *nouns*, but as these categories suggest, their system for cross-cultural communication is not developed for people with an intellectual disability (ID).

Takasaki & Mori (2007) (see also section 11.1.2) surveyed existing classification systems tailored to general audiences and children, and found that a categorisation system geared toward (non-disabled) children should prioritise popular topics, such as *entertainment* and *food*. Rather than using a standardised categorisation system, the authors take a more empirical method and build an interface with the help of educators, usability researchers, and pictograph designers. They propose the following categories: *basic* (i.e., frequently used) pictographs, *feelings*, *entertainment*, *people*, *places*, *relations* (i.e., adjectives and adverbs), *food*, *actions*, and *encyclopedia* (i.e., pictographs that are related to Takasaki & Mori's research project). Pictographs with a clearly defined relationship are placed next to each other. The longer the children used the tool, the more comfortable they became with the categorisation method, and ultimately better at finding pictographs. Note that this system is, again, not created

[1]Previous versions of section 8.1 appeared in Sevens et al. (2017a).

for people with ID.

In the Visual Inter Lingua (VIL) system by Leemans (2001) (see also section 11.1.3), the pictograph hierarchy has up to five levels of categories, with each non-terminal level containing between seven and nine subcategories. A hierarchy that has five levels and eight objects per level, can hold up to 32,768 (8^5) items. Note that VIL is designed as a case study on the topic of cross-cultural communication and that the hierarchy, in practice, only contains a handful of pictographs. In our approach, we will not go deeper than three levels, as we are targetting people with ID, who previously used a two-level hierarchy (see section 8.1.3). The pictographs in VIL are divided into *nouns*, *adjectives*, and *verbs*. For nouns, Leemans distinguishes between *physical world*, *beliefs and customs*, *arts and entertainment*, *sports*, *communication*, *science and technology*, and *transportation*. The adjectives are subdivided into *lower level perceptual*, *higher level evaluative,* and *comparative and relational* adjectives. Finally, for verbs, there are pictographs expressing *state and state changes* (with five subcategories) and *activities* (with two subcategories). Literature on the concept of prototypicality and semantics extensively supports these choices. Furthermore, Leemans recognises that some cross-reference of pictographs is necessary, as we shall also claim in our approach.

8.1.2 User-Centered Design

The static pictograph hierarchy is developed according to the principles of user-centered design. ISO (2010) sets out the four major user-centered design activities that are carried out in designing an interactive system:[2]

1. Understanding and specifying the context of use
2. Specifying the user requirements
3. Producing design solutions to meet these requirements
4. Evaluating the design against the requirements

These can be summarised as **analysis**, **specification**, **design**, and **evaluation**. The ISO standard acknowledges that these activities are not presented as waterfall steps; in real life, they must be carried out iteratively. User-centered design is a continuous

[2] https://thestandardinteractiondesignprocess.wordpress.com/introduction-2/

process where an understanding of what is needed is constantly negotiated between the designer, the commissioner, and the eventual users. In the ISO standard, each user-centered design activity implies a collection of sub-activities to be carried out to elicit and produce information. These are not specified in detail, and tend to vary between projects.

Work on the static pictograph interface, in particular, is largely done within the framework of the Able to Include project (see section 2.2.4.1). The first two steps in user-centered design, namely understanding and specifying the context of use and specifying the user requirements, were carried out by Daems et al. (2015). Daems et al. showed that communication on social media and understanding text is a central issue for people with ID. The authors also reported that users often find it hard to communicate with their friends through chatting, mailing, or texting, because the receiver might not understand the "language" the sender uses.

In the following sections, we discuss the next two phases, namely producing design solutions to meet the requirements (section 8.1.3), and evaluating the design with the end users (section 8.1.4).

8.1.3 Building a New Pictograph Interface

Before the improved pictograph translation technologies were introduced on the platform, WAI-NOT's pictograph interface (see section 2.1) contained a total of 2,479 Beta pictographs and 2,878 Sclera pictographs, which were spread across 21 and 39 different categories, respectively. The hierarchy was a two-level category system: All categories were shown on the first level, and an average of 118 (Beta) and 73 (Sclera) selectable pictographs were displayed on the second level. The positions of the pictographs were not fixed, but would change dynamically based on their frequency of use across all registered users. This made it hard for users to familiarise themselves with the locations of the pictographs. Because of this "frequency of use"-based system, concepts that are closely related were not shown together. For example, the *car* pictograph and the *tire* pictograph were separated by several dozens of unrelated pictographs. Furthermore, pictographs were not distributed across categories in a consistent way. For instance, the pictograph for *female teacher* could be found within the *school* category, but not within the *professions* or *people* categories. The pictograph for *male teacher* could be found within the *professions* category, but not within the *school* or *people* categories. Given the very large amount of pictographs per category, the overall inconsistency, the "frequency of use"-based mechanism, and the fact that no users were involved during the development and evaluation of the interface, we decided to adopt a different approach and to create a new pictograph hierarchy from scratch. We are using the WAI-NOT corpus (see section 2.2.4.2) to tailor this interface toward the end users' needs, interests, and vocabulary.

The new pictograph interface is a three-level category system (see Appendix F). For both Beta and Sclera, there are 12 top categories, which consist of 3 to 12 subcategories each. A total of 1,660 Beta pictographs (274 of which appear in multiple categories) and 2,181 Sclera pictographs (354 of which appear in multiple categories) are included,[3] and an average of 21 (for Beta) and 28 (for Sclera) pictographs can be found within each subcategory. All top-level categories have an orange border, while their subcategories have a light blue border. The pictographs that are located at the third level do not have a coloured border; these are the ones that can be selected by a user to create messages. When navigating through the hierarchy, a user can move back to a higher level by tapping the *back* arrow, which shares the colour of its border with the upper-level pictographs. A small navigation panel is provided at all times, showing the user's current location inside the pictograph hierarchy (see Figure 8.1). Mixed input (consisting of both pictographs and text) is also allowed.

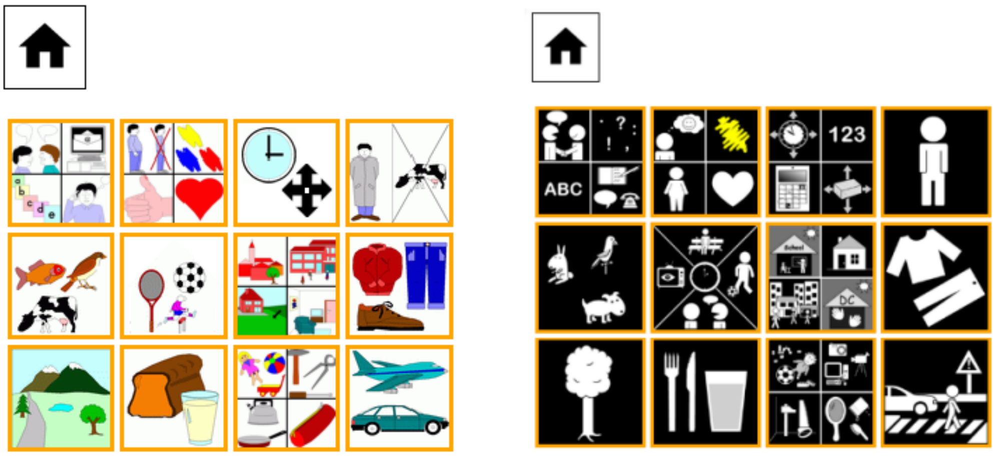


Figure 8.1: The top-level (*Home*) categories in the Beta interface (left) and the Sclera interface (right). The top-level pictographs are assigned an orange border, indicating that a user can click on them in order to access the subcategories.

[3] This is less than the baseline, since pictographs that were not used in the baseline system are removed.

Our choice for the 12 **top-level categories** is motivated by the results of a Latent Dirichlet Allocation (LDA) analysis with the MALLET toolkit (McCallum 2002) applied to the WAI-NOT corpus, using standard settings.[4] LDA is a technique that allows to determine prominent hidden topics, i.e., the things that our users like to talk about. To create these hidden topics, MALLET clusters words that frequently occur together. For example, one prominent hidden topic that is retrieved by MALLET is made up of the following words (translated from Dutch): *fries*, *hamburger*, *waffle*, *pizza*, *croissant*, *drink*, *delicious*, *salt*, etc. It can be inferred that all these words relate to the domain of food and drinks. Based on the hidden topics that are retrieved, we create the following categories: *conversation*, *feelings & behaviour*, *dimensions*, *people*, *animals*, *leisure*, *locations*, *clothing*, *nature*, *food & drinks*, *objects*, and *traffic & vehicles*. These 12 top-level categories can be placed in a 3x4-sized grid, making this hierarchy also more appropriate for people who use specialised input devices, since less movements are needed to access each category. Note that some categories, like *locations* or *objects*, are difficult to represent with one pictograph. For these categories, we created a new pictograph, which shows four representative members of that category.

The **subcategories** were largely formed by exploring Cornetto's (Vossen et al. 2008) relations between concepts (see section 2.3.1.1), namely the hyperonymy (i.e., a more general concept) and hyponymy (i.e., a more specific concept) relations. For example, the following subcategories for the *animal* category were formed: *dogs*, *cats*, *rodents*, *birds*, *aquatic animals*, *farm animals*, *insects*, and *wild animals*.

Finally, pictographs occurring **within each subcategory**, which can be selected by the user to compose pictograph messages, are assigned manually (see Figure 8.2). We order them in accordance with the frequency with which they appear in the WAI-NOT email corpus (in other words, the frequency with which they were used in WAI-NOT's baseline version), although there are some exceptions. Examples of the latter are the natural ordering of numbers and months (*1*, *2*, *3*,...), pairs of antonyms (*big* and *small*), or concepts that are closely related.[5] For example, *to brush your teeth* is placed next to the *toothbrush* and *toothpaste* pictographs. Note that some pictographs may appear in different subcategories. For instance, the *teacher* pictograph appears in the *people > professions* category and in the *locations > school* category, making it easier for the users to find. In a handful of cases, the third level contains a link to another

[4]We adopted this approach, rather than using pre-established categories, to take the characteristics of our target group into account.

[5]This is in a way similar to semantic frames (Fillmore 1982).

second-level category. For instance, there is a direct connection to the *food > cooking & baking* category inside the *locations > kitchen* category. In the new interface, the pictographs' locations are fixed and they do no longer change based on their frequency of use.

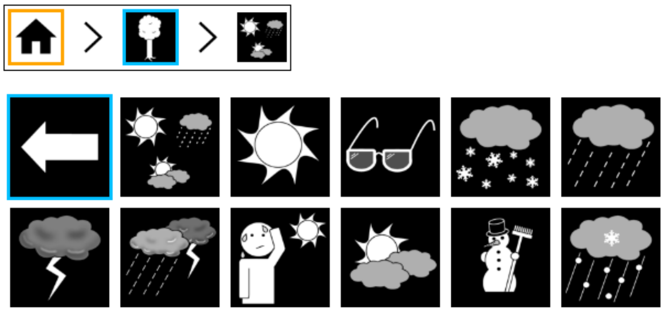


Figure 8.2: *Home > Nature > Weather*. This category contains pictographs that are related to weather. Clicking the orange-bordered *home* pictograph takes the user back to the top level, while the blue-bordered pictographs link back to the second level in the hierarchy.

In order to confirm the (potential) success of this approach, we made a comparison between 1,000 randomly selected messages sent with the old WAI-NOT interface and 1,000 randomly selected messages sent with the new interface (i.e., emails that were sent after February 2017, when the old interface was replaced by the new interface). In the old system, 16.6% of all messages were pictograph-based (i.e., they did not contain any form of written text), whereas 25% of all messages are pictograph-based in the new system. This observation could indicate that people who depend on pictograph input find the upgraded hierarchy more accessible than the old interface. Furthermore, we found that 68% of the new pictograph-based emails have a clear communicative message[6] (versus 30% in the old system), with merely 15% of the messages being random and meaningless (versus 52% in the old system).[7] In 74% of the cases, the users of the new system explored other subcategories when creating their messages (versus 43% in the old system). However, note that subcategories in

[6]We judged whether the combined meaning of the pictographs made sense (i.e., whether the conveyed message was likely to have been communicated intentionally by the user, or whether the message was, rather, the result of random clicking). In the case of random clicking, we were not able to find a meaningful connection between the pictographs. Note that these messages usually contain pictographs that belong to the same category, that appear closely together in the interface.

[7]There is a grey zone of 17% in the new system, and 19% in the old system, as it is unclear to us whether these messages are random or truly meaningful.

the old hierarchy contained many more pictographs. Therefore, moving to different subcategories might not have always been necessary.

8.1.4 User Studies

The new static hierarchy was tested and discussed with the end users through observation and semi-structured interviews in focus groups (see section 2.2.4.1). The focus group consists of five young adults with ID, one female user and four male users, from a day centre and community in Belgium. Three out of five participants have an additional motor disability. The users' needs and expectations were revealed by means of group discussions and semi-structured interviews. For testing, a WAI-NOT account was set up and the five participants were asked to create short messages or to identify pictographs using the new hierarchy. Individual assistance was given by one of their caregivers, who would clarify some of the questions to the users.

We prepared a number of questions, which we used as a guideline for our conversations with the target users and their coach during the hands-on session:

- Why do you use social media websites and email?
- What do you like to talk about on social media websites and email?
- Would you like to use pictographs to communicate? Why?
- Do you think navigation in the hierarchy is easy? Can you find some of your favourite pictographs? (For example, favourite hobby, favourite drink, etc.)
- According to you, are there too many pictographs in the hierarchy?

The participants primarily use (or wish to use) social media websites in order to stay in touch with their friends, family, and people that they don't see often. When chatting, users like to ask how their conversational partner is doing, talk about their daily activities, and make plans to meet up. However, since some of the participants need personal assistance from a caregiver to write messages, due to their motor disability and/or limited writing skills, private matters are not always discussed online. The three users who did not yet write messages independently admitted that they would chat about different topics if their coaches were not present, such as about being in love, or gossip about their caregivers. It should also be noted that individual assistance during social media use demands an additional effort from the caregivers.

One of the participants had previous experience with "AbleChat" (Daems et al. 2015), an app designed within the framework of the Able to Include project (see section 2.2.4.1), which provides the user with a very small, fully customisable set of pictographs that can be used for very basic communication needs, primarily within the context of independent mobility (including, for example, pictographs for greetings, traffic, and requests for assistance). While the participant admitted that she highly enjoyed using the pictograph-based chat tool and acknowledged that the app made independent communication easier - or even possible - for her, she also explained that the pictograph set in AbleChat was too small to discuss everyday activities and things. The idea of being able to choose from a large set of pictographs, covering a wide range of phenomena, and thus allowing the user to write about virtually anything, strongly appealed to all participants.

Each of the participants tested the new pictograph interface during a hands-on session. All five participants intuitively understood that the *back* arrow could be used in order to move back to a higher level in the interface, indicating that the coloured borders made navigation obvious. No further instructions were needed. One of the users suggested adding an *enlarge pictographs* setting for people with a visual impairment, like himself.

The users were prompted to find a pictograph of their choice. Even without previous knowledge of pictograph languages, they did not experience major difficulties in locating their desired target image, although we expect frequent use and familiarisation with the interface to have a positive effect on pictograph retrieval efficiency over time. While the large amount of subcategories may be confusing for some users, the participants stated their will to learn how to work with them. They also underlined the importance of having access to all subcategories at all times. After all, it could be the case that the user would like to express that, for instance, his/her brother or sister is competing in an important football match, even if that particular user does not like sports himself/herself at all. One participant, who had previous experience with the old WAI-NOT email system, confirmed that the new interface was less confusing.

The participants were enthusiastic about having access to a large pictograph set and claimed that they would use the Pictograph-to-Text translation system on social media websites.

8.1.5 Conclusion: A Static Hierarchy

The new interface is, overall, a major improvement over the old category system. Although our five participants agreed that all (sub)categories must be present in their interface at all times, allowing them to communicate about a wide range of topics, it is imaginable that the hierarchy can be too overwhelming for some users - especially for beginning pictograph users. Therefore, we enable the parent or caregiver to disable or activate specific (sub)categories, in order to personalise a user's pictograph interface based on his/her individual interests. By default, all categories are enabled, but one may choose to disable certain (sub)categories because they are not appropriate or too confusing for a user. Further testing will be necessary in order to confirm the validity of these new customisation settings.

8.2 A Dynamic Pictograph Prediction System

The second input method is a dynamic pictograph prediction tool. It suggests contextually and/or semantically relevant pictographs, based on the pictographs that were previously selected by the user.[8] We first present related work on the topic of word prediction and its implications for our prediction task (section 8.2.1). We then describe two approaches toward pictograph prediction. The first approach is based on *n*-grams (section 8.2.2), while the second approach relies on semantic associations (section 8.2.3). Using a test set of user-generated pictograph messages, we evaluate both tools and the combination of *n*-gram probabilities and semantic knowledge (section 8.2.4). Finally, we conclude (section 8.2.5).

8.2.1 Status Quæstionis: Word Prediction

Word prediction is the task of predicting what a user is going to type, as he/she is typing (Stoop & van den Bosch 2014:2). A well-known application of this technology

[8]Note that the pictograph prediction tool can only use contextual information to generate suggestions, as opposed to most word prediction tools, which do not only *suggest* the next word based on the previously entered words, but also *predict* the current word by using the first characters of that word. In this regard, when using the term "pictograph predictions", we are referring to "hypothetical" predictions that serve as *suggestions* to the user, as there is no further information available in order to make informed decisions.

is Google's autocomplete function,[9] which helps people find the information they are looking for faster using search predictions. Word prediction technology reduces the number of keystrokes a user has to make, thus saving time and preventing mistakes, and can often be found on digital devices, such as smartphones or tablets. Before 2000, when mobile phones were not yet widely used, most studies on predictive editing targeted disabled users, aiming to reduce their effort spent on entering text.

The earliest word prediction models made use of large frequency lists. Swiffin et al. (1985)'s prediction tool, for instance, suggests a number of high-frequency words that match the partially typed word, but ignores the context of that word. Later, *n*-gram models improved prediction accuracy further by taking the context of previously entered words into account and making use of language models. However, as noted by Stoop & van den Bosch (2014:5), the accuracy of context-sensitive prediction systems largely depends on how often a similar context is available in the training material. Standard *n*-gram models are often augmented with linguistic knowledge, such as part-of-speech tags.

Most recent word prediction systems include semantic information with the objective of improving prediction accuracy. Semantics-based methods help to model more of what is not captured by *n*-grams, namely long-distance co-occurrence relations between words. Stocky et al. (2004), for instance, use commonsense knowledge to generate words that are semantically related to the user's input, using a large database of semantic relationships, but they do not use *n*-gram information. Similarly, Wiegand & Patel (2012) present a prediction tool that does not require grammatical input during either training or testing, arguing that selecting words serially and in syntactic order can be physically and cognitively arduous for users with a disability. Since people with a disability often produce syntactically incomplete or incorrect input, Wiegand & Patel implement prediction at the word level using sem-grams, which provide relational information between different segments of the text.

Li & Hirst (2005) combine semantic knowledge with *n*-gram probabilities to predict words that are semantically more appropriate than *n*-grams alone, arguing that human language processing also involves semantics: While reading through a text, people may predict upcoming words by using previous semantic information from the text. Li & Hirst propose a prediction model in which semantic information is integrated with an *n*-gram model. More specifically, the predictions of the *n*-gram model are filtered and re-arranged by the semantic model.

[9] https://www.google.com/

An important factor that should be taken into account is the cognitive and physical (i.e., movement-related) load that the prediction system imposes on a user (Trnka et al. 2016:20), including the focus of attention shifts from the keyboard to the prediction list, the scanning of the list, and deciding whether the desired word (or pictograph) appears in the list. The way in which a user's communication rate is enhanced, depends largely on the accuracy of the prediction system. Users may find poor predictions more distracting than actually helpful. On the other hand, if a user trusts the system more, he/she will scan the prediction list more often. Trnka et al. (2016:23) conclude that the cost of added cognitive load is often outweighed by the benefit of the keystroke savings offered by the prediction engine.

One important difference between building a prediction system for pictograph communication and building a prediction system for textual communication is the fact that pictographs are selected with one click or tap, while words consist of characters, which may aid in gradually narrowing down the suggestion list. In other words, the only information that our system has at its disposal, are the pictographs that were previously entered by the user within the context of that same message.[10] Furthermore, creating a language model for context-sensitive pictograph prediction is not a trivial task, as we do not have a large corpus of pictograph-based messages at our disposal.

To our knowledge, there exists only one study that has previously described and evaluated the performance of a pictograph prediction system. García et al. (2015) propose a prediction tool that makes use of statistical language models (up to bigrams). The pictograph corpus used to train these language models was handcrafted by twelve non-disabled undergraduate students, who were asked to use the available pictographs (belonging to a 2-level hierarchy with 19 categories) to produce sentences that could be used in three different contexts: the classroom, a cafeteria, and home. The students created an average of ten pictograph sentences per location, and they did not construct sentences that contained more than five pictographs. Note that this pictograph corpus is very small and does not contain real conversation material of people with a disability. The authors note that this is a common limitation

[10] In section 5.3.1, we highlighted some of the similarities between the Chinese writing system and Sclera and Beta pictographs. When typing, Chinese users enter pronunciations, which are converted into relevant Chinese characters. The user must select the desired character from a list of homophones. They are aided by modern systems, such as Google Pinyin, which predict characters based on context and user preferences. This method is not appropriate for our users.

in vocabulary prediction studies, since most of the times, the corpora used are based on written or spoken communication from non-disabled persons. Furthermore, their simulations, which are used to evaluate different versions of the prediction system, assume an "optimal user" behaviour concerning use of pictograph prediction.

8.2.2 Predictions Based on *n*-gram Information

Our first prediction model relies on the pictograph's immediate context or *n*-gram information. We automatically translated the complete WAI-NOT email corpus (see section 2.2.4.2) into pictographs using the baseline Text-to-Pictograph translation system (see Chapter 3). After having deleted all words that did not correspond to a valid pictograph file name, we ended up with a Sclera corpus consisting of 259,854 pictographs and a Beta corpus consisting of 242,653 pictographs. We built two language models, one for each pictograph language, using the SRILM toolkit (Stolcke 2002), using standard settings. These language models contain unigram, bigram, and trigram information on pictograph sequences.

When the user has not yet inserted any pictographs, the prediction tool uses the language model to look up bigram information for "*<s>* (sentence beginning) + any new pictograph". In this case, using the probability information that is stored in the language model, the tool suggests the pictographs that are connected to the concepts of *hello*, *I*, *good*, *you*, *and*, and so on, as these concepts are most commonly used by our target users to commence their messages.

When at least one pictograph has been inserted by the user, either with the help of the prediction tool or by browsing the static hierarchy, the system uses the previous two pictographs (or the sentence beginning and the first pictograph) to look up trigram information in the language model. If no (or not enough) trigrams are found, the system backs off to a lower-order *n*-gram using Katz's back-off model, until a sufficiently large amount of pictographs is retrieved. The total desired amount of suggestions can be set by means of a parameter.

The highest scoring *n*-grams are used to present context-sensitive pictograph predictions to the user. For example, if the user inserts the Beta pictographs *komen.png* 'come', *jij.png* 'you', and *naar.png* 'to' (corresponding to the Dutch phrase *kom jij naar* 'are you coming to'), the system will predict the following pictographs: *school.png* 'school', *feest.png* '(the) party', *mijn.png* 'my', *mij.png* 'me', *ons.png* 'our', *buiten.png* 'outside', and so on.

8.2.3 Predictions Based on Semantic Associations

Our second prediction model relies on word associations within a context that is broader than *n*. For each pictograph that has been inserted by the user, the system identifies the most frequent lemma in its synset. This lemma is used to generate a list of semantically similar words from DISCO (Kolb 2008),[11] a tool that retrieves the semantic similarity between arbitrary words and phrases, along with their similarity scores. After having obtained such a list of semantically similar words (i.e., lemmas) for every pictograph, we calculate the overall similarity scores for the complete pictograph string.

The prediction system queries the Cornetto database (see section 2.3.1.1) to determine whether there are any pictographs connected to the suggested lemmas. Note that, if more than one pictograph is found for a lemma, the most frequent sense of its connected synset is chosen.

In contrast with the *n*-gram model, which only uses the two previously selected pictographs, the semantic association model uses the complete pictograph string to suggest semantically relevant, but not necessarily syntactically relevant, pictographs to the user. For instance, if the user inserts the Beta pictographs *lieveheersbeestje.png* 'ladybug', *mier.png* 'ant', and *mug.png* 'mosquito', the system will predict the following pictographs: *kever.png* 'beetle', *wesp.png* 'wasp', *vlieg.png* 'fly', and so on. Note that the semantic model cannot generate pictographs for non-content words, such as pronouns or conjunctions, as these parts-of-speech are not retrieved by the DISCO tool.

8.2.4 Evaluation of the Prediction Tools

We randomly selected 50 pictograph-based emails (corresponding to a total of 273 pictographs) that were sent by WAI-NOT users using the improved static pictograph hierarchy, which we deemed to have a clear communicative message (see section 8.1.3). If the message was written in Beta pictographs, we manually translated it into Sclera pictographs, and vice versa. Note that these emails did not appear in the WAI-NOT corpus which we used for training the *n*-gram prediction model.

The objective of this evaluation is to determine whether each of these 273 pictographs can be predicted accurately by the prediction models, based on the previously entered pictographs (or the sentence beginning). If the correct pictograph

[11] http://www.linguatools.de/disco/

	Beta		Sclera	
# Predictions	Recall	Avg. position	Recall	Avg. position
***n*-gram model**				
1000	88.64%	181	86.81%	205
200	62.64%	52	58.24%	48
100	49.08%	27	46.15%	24
50	37.36%	15	35.90%	11
10	17.58%	3	21.61%	4
Association model				
1000	26.73%	82	20.88%	67
200	13.92%	27	12.09%	22
100	9.89%	16	9.89&	12
50	7.69%	10	8.06%	9
10	3.66%	3	3.66%	3
Combined model				
1000	88.64%	178	86.81%	202
200	62.64%	49	58.24%	46
100	49.08%	25	46.15%	23
50	37.36%	14	35.90%	11
10	17.58%	3	21.61%	4

Table 8.2: Recall of the pictographs to be predicted according to the different models and their average position within the suggestion list, for different sizes of suggestion lists.

is predicted, we also calculate its position within the suggestion list. We show the results for generating up to 10, 50, 100, 200, and 1000 predictions.

The results are presented in Table 8.2. Looking at the basic prediction models, it can be seen that the *n*-gram model outperforms the association model, both in terms of **recall** and **average position** of the target pictograph within the suggestion list. We dug deeper and found that using only semantic information often does not allow the prediction tool to make informed decisions about the remainder of the message. For instance, the pictograph sequence *zaterdag.png* 'Saturday' and *ik.png* 'I' lead the semantic prediction model to generate a number of pictographs that represent the other days of the week. However, it is not able to propose activities, like *zwemmen.png* 'to swim', as these pictographs are not semantically related to any of the two previously entered pictographs.

Table 8.2 shows a clear trade-off between the recall and the average position of the pictographs to be predicted for different sizes of suggestion lists. Large suggestion lists have higher recall, but cause the average position of the pictographs to be predicted to be lower (i.e., located further away from first place). Conversely, small suggestion lists have lower recall, but cause the average position of the pictographs to be predicted to be higher. In real-time applications, like email clients, a suggestion list size of 10 is acceptable. Using the *n*-gram model, a Beta user should be able to find his/her desired pictograph in the suggestion list in approximately 17.58% of the cases, whereas a Sclera user should be able to find it in approximately 21.61% of the cases. This implies that the user would not have to look for the target pictograph in the static hierarchy in approximately one out of five cases. The average position of the pictograph in the suggestion list is 3 for Beta, and 4 for Sclera.

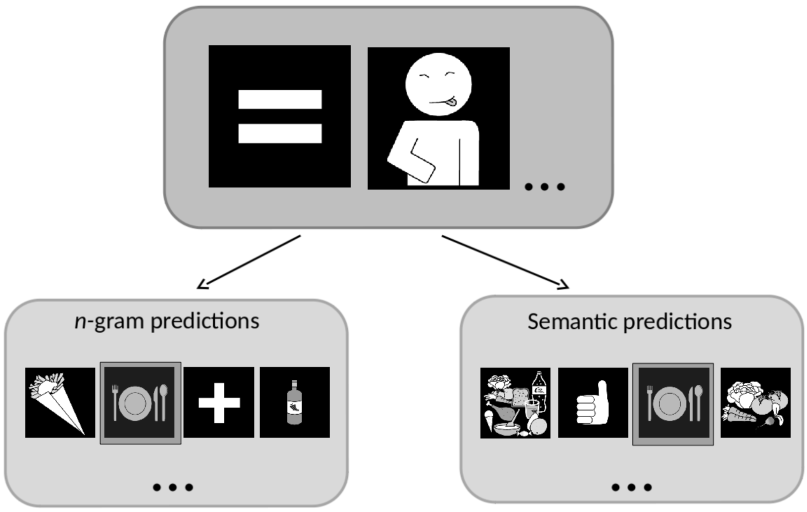


Figure 8.3: The user inserts the Sclera pictographs *gelijkheidsteken.png* 'to be' and *lekker.png* 'delicious' (corresponding to the Dutch phrase *(het) was lekker/(het) waren lekkere* '[...] was/were delicious' or 'it was delicious'). The *n*-gram model suggests *frietjes.png* 'fries', *eten.png* 'food', *plus.png* 'and', and *wijn.png* 'wijn', among other suggestions. The semantic association model suggests *voedingsmiddelen.png* 'foodstuffs', *duim-omhoog.png* 'nice', *eten.png* 'food', and *groenten.png* 'vegetables', among other suggestions. Observe that *eten.png* 'food' appears in both types of suggestion lists.

We also verified whether the two models can be combined for more accurate predictions (see Figure 8.3). The above-mentioned tools can be merged in (at least) two different ways. The first combined model uses the *n*-gram model as its "core" prediction module. As discussed before, the n-gram model does not take the semantics of the whole sequence into account, but only uses the two previously inserted pictographs. The combined model re-ranks the list of pictographs as suggested by the n-gram model using the predictions made by the semantic association model, thus augmenting the context-based predictor with semantic information. For every n-gram prediction, if an identical prediction is found in the association model's suggestion list, its positional score (i.e., for n predictions generated, a score of n for the pictograph with the highest log probability, down to a score of 1 for the pictograph with the lowest log probability) is augmented (i.e., added up) with the positional score of that same pictograph as suggested by the association model. Note that pictographs that are suggested by the association model, but not by the n-gram model, are by default excluded from the combined suggestion list. The results are presented in Table 8.2. The second combined model works the other way around and uses the semantic association model as its "core" prediction module. We do not consider this approach, as the recall of the semantic association is too low, and re-ranking its output would not necessarily lead to useful results in real-time applications. Table 8.2 shows that combining the models does not lead to improvements on small suggestion lists; improvement is not significant on large suggestion lists, either. Again, this can be attributed to the fact that the messages are short and that there is often not enough semantic information available for the tool in order for the semantic association model to be truly useful.

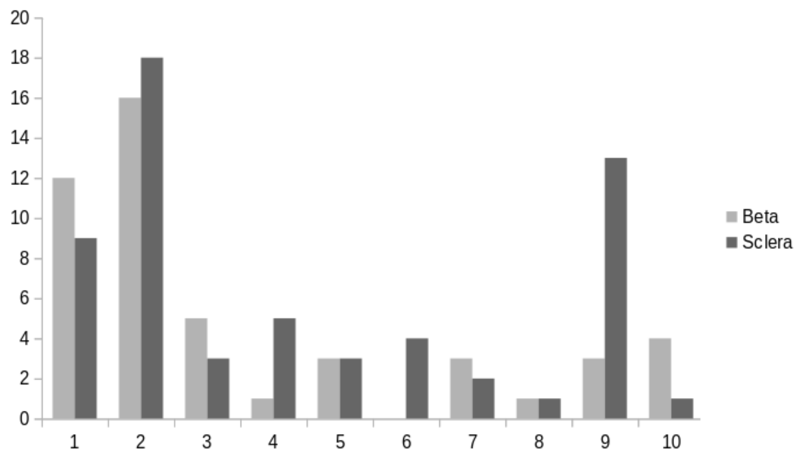


Figure 8.4: Positional distribution of the predicted pictographs across ranks in a suggestion list of size 10.

For a suggestion list of size 10, and using the regular *n*-gram prediction model, Figure 8.4 shows the positional distribution of the pictographs that were accurately predicted (with first place being the highest ranked suggestion). Out of the 273 pictographs in our test set, a total of 49 Beta pictographs (17.58%) and 59 Sclera pictographs (21.61%) were predicted correctly. For Beta, 67.35% of these pictographs appear in top 3 position, whereas for Sclera, 50.85% of these pictographs appear in top 3 position.

Persona: Vivi (After)
Freya installed the pictograph interface on Vivi's smartphone, introducing a limited set of Beta pictographs that relate to feelings, traffic, locations, and emergencies. She will enable more pictograph categories in the future, but for the time being, only relevant images are shown. Freya feels more relieved, knowing that Vivi can now travel home independently. One day, Vivi sends her mother the pictograph of a snack bar, followed by three smiley faces. She probably won't be home for dinner...

8.2.5 Conclusion: A Dynamic Prediction System

The performance of the prediction models, and especially the *n*-gram prediction model, can be considered satisfactory, given that the context of the messages is usually rather narrow (a message consists of 7.7 words or pictographs on the average, see section 2.2.4.2), and given that several thousands of pictographs are available in the translation system for a user to choose from. Furthermore, the system often has to make uninformed decisions. For instance, a user could be eating virtually *anything* (so which food pictographs will the system suggest?), go to the swimming pool on *any* day of the week (so what will be the order in which the days of the week are suggested by the system?), and so on. One way to avoid this would be by training the prediction models on individual users and inferring their personal preferences over time. For instance, the model could learn that a user likes to eat pancakes, or that he/she usually goes to the swimming pool on Wednesdays. Implementing such a system would require us to log the input of each user, and (continuously) update the language model according to his/her characteristic pictograph use. While user-based logging (and its related privacy issues) is beyond the scope of this dissertation, it is the logical next step in the development of a truly intelligent prediction system.

CHAPTER 9

Development of a Pictograph-to-Text Translation Tool

The Pictograph-to-Text translation tool translates pictographs (or a combination of written text and pictographs) into natural language text. Its objective is to facilitate the construction of written content.

After a brief discussion of related work (section 9.1), we present and evaluate the baseline system, which generates text from pictographs using WordNet synsets and n-gram or long short-term memory (LSTM) language models (section 9.2), and we assess the effects of using different types of corpora for the language modelling task. Next, we evaluate how purely rule-based methods (section 9.3) and data-driven machine translation approaches (section 9.4) could make up for the shortcomings of the baseline system. We will argue that the contributions of the rule-based approach are too small to compete with most contemporary data-driven machine translation methods, since the recall of these systems is too low and their development is particularly labour-intensive. More promising results can be obtained, however, using data-driven machine translation approaches. This chapter contributes to the ongoing debate on whether neural machine translation systems are able to outperform phrase-based systems, and presents the advantages and disadvantages of using either method. We rate and rank the output of the n-gram language model, the LSTM language model, the phrase-based machine translation system, and the neural machine translation system and verify whether human judgments correlate with the output of automated metrics (section 9.5). Finally, we present our conclusions (section 9.6).

Persona: Chara (Before)
Chara confuses similar words and she often has trouble coming up with the specific vocabulary to express her ideas or thoughts, making it difficult for her to get her message across. This frustrates Chara beyond measure. In her daily life, she sometimes uses pictograph cards to communicate her feelings or needs. In online environments, however, she does not have access to these cards. She often feels insecure and she does not write a lot of messages online.

9.1 Status Quæstionis: Generating Text from Labelled Images

Our baseline translation model uses ideas from early work on natural language generation. Jing (1998) retrieves semantic concepts from WordNet and produces lexical paraphrases for a specific application domain. She demonstrates that WordNet is a valuable resource for generation, as it can produce large amounts of paraphrases, provide a semantic net (i.e., it can map the semantic concepts to be conveyed to appropriate words) for lexicalisation, and can be used for building domain ontologies. As there may exist multiple surface forms for a concept, Liu et al. (2003) use statistical language models as a solution to the word inflection problem. A language model re-scores the inflected forms and generates the most probable hypothesis.

There exist a number of translation engines that automatically generate text from images. For a more detailed discussion of their architecture and their shortcomings, we refer to Chapter 11, in which we compare our technologies with other pictograph-based translation systems. Finch et al. (2011) developed picoTrans, a mobile application which allows users to build a source text by combining pictures or common phrases, but their application is not intended for people with a disability. The Prothèse Vocale Intelligente (PVI) system by Vaillant (1998) offers a limited vocabulary of pictographs, which correspond to single words. The PVI system searches for predicative elements, such as verbs, and attempts to fill in its semantic slots. When all slots are filled, a grammatical sentence can be generated. Fitrianie & Rothkrantz (2006) apply a similar method, requiring the user to first select the pictograph of the verb and then fill in the role slots made available by that verb. Various pictograph chat applications, such as Messenger Visual (Tuset et al. 2010) and Pictograph Chat Communicator III (Munemori et al. 2010), allow the user to insert pictographs, but they do not generate natural language text.

The Pictograph-to-Text translation engine differs from these applications in that it is specifically designed for people with an intellectual disability, does not impose any limits on the way or order in which the pictographs are inserted by the user, and always generates natural language output (that is grammatical, where possible).

9.2 The Baseline System: Language Models

The baseline system for Pictograph-to-Text translation generates natural language from pictographs using language models, without using grammatical information in the translation process.[1] We first discuss the system's architecture. It generates lexical surface forms for the pictographs that are inserted by the user (section 9.2.1) and calculates an optimal natural language sequence (section 9.2.2). We then evaluate the baseline system (section 9.2.3) and describe its shortcomings (section 9.2.4).

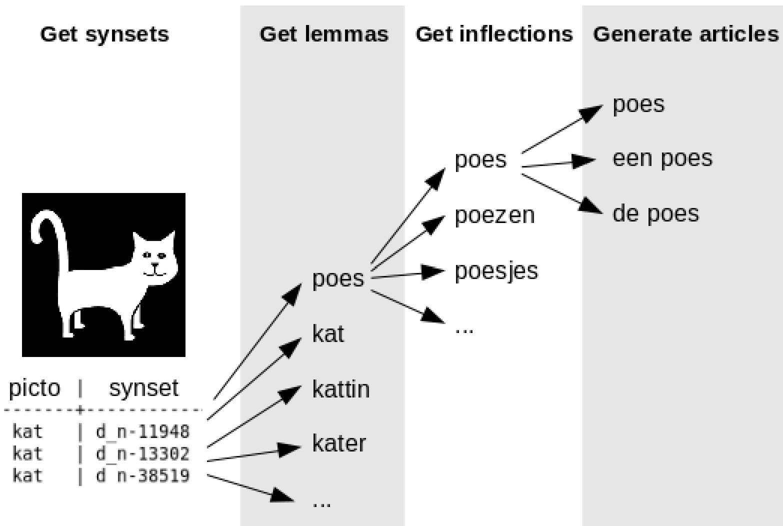


Figure 9.1: An example of surface form generation.

[1]Previous versions of sections 9.2.1 to 9.2.4 appeared in Sevens et al. (2015b). The system described in Sevens et al. (2015b) has the same architecture as the n-gram-based approach that is discussed in the upcoming sections, but it uses trigram language models on a small mixed-domain corpus, instead of 5-gram language models or LSTM language models. We also evaluate the system with different corpora, including a larger mixed-domain corpus.

9.2.1 Generating Surface Forms for Each Pictograph

When a pictograph is selected, its Cornetto synset is retrieved (see section 2.3.1.1), and from this synset, the system retrieves all the synonyms it contains. For each of these synonyms, *reverse lemmatisation* is applied. The reverse lemmatiser retrieves the full, inflected linguistic paradigm of the lemma, along with its part-of-speech tags. The reverse lemmatiser that we use in our system is based on the SoNaR corpus (Oostdijk et al. 2013).

Each of these surface forms is a hypothesis for the language model, as described in section 9.2.2. For nouns, the system generates additional alternative hypotheses which include an (indefinite or definite) article, based on part-of-speech information (see Figure 9.1).[2]

9.2.2 Decoding

The goal of statistical language modelling is to learn the joint probability function of sequences of words in a language (Bengio et al. 2003:1137).

Traditional approaches based on *n*-grams obtain generalisations by concatenating very short overlapping sequences seen in the training set. Section 9.2.2.1 describes a **beam search decoder** that uses **5-gram language models** to calculate an optimal natural language translation for a pictograph input string.

Long short-term memory-based (LSTM) language models are language models that are based on recurrent neural networks. We describe our approach toward LSTM language modelling for Pictograph-to-Text translation in section 9.2.2.2.

Both types of language models are trained on **three different corpora**. The first training corpus is a large mixed-domain corpus consisting of the resources that are shown in Table 9.2. We also train our language models on two smaller corpora, which contain domain-specific data. One is the Flemish part of Corpus Gesproken Nederlands (CGN) (Oostdijk et al. 2002) (3.8M tokens). This corpus contains transcriptions of spoken language, which, in many ways, resembles spontaneous chat

[2]Note that this generation module has its limits, and could be improved. For instance, while the system is currently able to generate ungrammatical adjective-article-noun sequences like *zwarte de kat* 'black the cat', it cannot yet generate grammatical article-adjective-noun sequences like *de zwarte kat* 'the black cat', since the generation module dictates that articles must appear directly in front of nouns. Furthermore, we should consider enabling the generation of more function words, such as prepositions. Adding more (robust) rules to the generation module is future work.

conversations in online environments. The other one is the "subtitles" component of the SoNaR corpus (Oostdijk et al. 2013) (27.6M tokens). Both corpora contain many first-person and second-person forms.

Name of resource	**Reference**	**Number of tokens**
Open Subtitles 2016	Lison & Tiedemann (2016)	266.6M
EUbookshop	Skadins et al. (2014)	245.0M
DGT	Tiedemann (2012)	72.6M
Europarl	Tiedemann (2012)	59.5M
Wikipedia	Tiedemann (2012)	19.7M
CGN (Flemish)	Oostdijk et al. (2002)	3.8M
SoNaR-500	Oostdijk et al. (2013)	500M

Table 9.2: The used resources in the mixed-domain corpus.

9.2.2.1 *n*-gram Models

In the *n*-gram-based approach, the system performs beam search decoding[3] on a 5-gram language model with interpolated modified Kneser-Ney smoothing (Chen & Goodman 1999),[4] trained with the KenLM Language Model Toolkit (Heafield et al. 2013) on the three corpora.

The 5-gram-based Pictograph-to-Text translation system contains a number of decoding parameters. *Threshold pruning* determines whether a new path should be added to the existing beam in the search space, based on the probability of that path as compared to the best path. *Histogram pruning* sets the beam width (in other words, only the *n* best hypotheses are kept). The *cost* parameter estimates the cost of the pictographs that still need processing, based on the amount of pictographs that have not yet been processed. Finally, the *reverse lemmatiser's minimum frequency* sets a threshold on the frequency of a token/part-of-speech/lemma combination in the corpus, and reduces the total amount of possible surface forms that are retrieved for a particular pictograph. These frequencies are based on occurrence within the SoNaR corpus (Oostdijk et al. 2013).

We tune these parameters for the Dutch-to-Sclera and Dutch-to-Beta translation engines. The tuning set consists of 50 manually translated messages from the WAI-

[3] Beam search is a heuristic search algorithm that explores a graph by expanding the most promising node first.

[4] A method that smooths language models by moving some probability toward unknown *n*-grams.

Parameter	Min	Max	Step	Sclera	Beta
Threshold pruning	0	20	1	3	4
Histogram pruning	0	20	1	4	12
Cost	0	20	1	2	2
Rev. lem. minimum frequency	2	41	1	36	21

Table 9.3: Parameter values for the baseline Pictograph-to-Text translation system after tuning.

NOT corpus. Note that the source sentences in this task correspond to the reference translations that were used for tuning the Text-to-Pictograph translation system (see Appendix B.1). We run five trials of local hill climbing (Vandeghinste et al. 2017) on the parameter search space, with random initialisation values, in order to maximise BLEU (Papineni et al. 2002). We vary the values between certain boundaries, and with a parameter step size of one, in order to cover different areas of the search space. For the reverse lemmatiser, we build 6 versions (i.e., reverse lemmatisers with a minimum frequency of occurrence per token of 2, 11, 21, 31, 36, and 41). We run each trial until BLEU converges onto a fixed score. From these trials, we take the optimal parameter settings (see Table 9.3).

9.2.2.2 Long Short-Term Memory Models

Neural networks can be used to extract patterns and detect trends that are too complex to be noticed by humans or other computer techniques (Stergiou & Siganos 2018). Instead of computing the output value directly from the input values, neural networks introduce hidden states.[5] Training neural networks requires the optimisation of weight values so that the network predicts the correct output for a set of training examples. The most common training method for neural networks is called back-propagation, since it first updates the weights to the output layer, and propagates back error information to earlier layers. These weight updates are driven by a gradient toward a smaller error.[6] Once the weights are updated, the next training example is processed. Typically, there are several passes over the training set, called

[5]They are called hidden, because we can observe inputs and outputs in training instances, but not the mechanism that connects them (Koehn 2017).

[6]In other words, the objective of the gradient descent algorithm is to find a set of weights and biases which make the cost as small as possible.

epochs. A schematic representation of a neural network is shown in Figure 9.2.

Traditional neural networks cannot use their reasoning about previous events to classify what kind of event is happening at every point (Olah 2015). Recurrent neural networks address this issue. They are networks with loops in them, allowing information to persist. However, it is possible for the gap between the relevant information and the point where it is needed to become very large. Long short-term memory (LSTM) networks are a special kind of recurrent neural network which are capable of handling such long-distance dependencies.

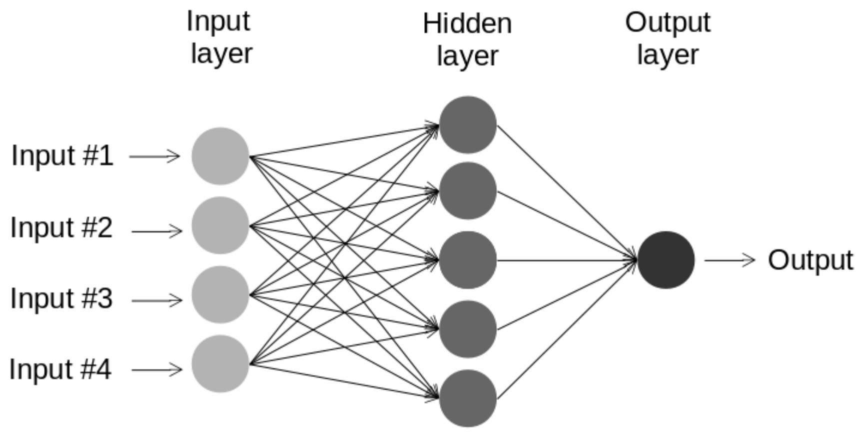


Figure 9.2: Schematic representation of a neural network.

For each of the three corpora, we build a bidirectional LSTM language model (BRNN) with 750 hidden nodes using the OpenNMT toolkit (Klein et al. 2017). The bidirectional encoder splits the neurons of a regular recurrent neural network into two directions: one for positive time direction, and another for negative time direction. The outputs of the two networks are summed at each time step. This structure allows the networks to have both backward and forward information about the sequence at every time step.

Each word is assigned to an index within the word and features vocabularies. In this case, we limit our word vocabulary to 100,000 tokens (i.e., the most frequent 100,000 words are treated as unique, and all other words are converted to an "unknown" token and get the same embedding). By default, OpenNMT shuffles and sorts the data before the training. During training, batches are selected at random.

We present all the relevant parameter values that are used during training and decoding in Table 9.4. Note that we used mostly default settings.

Parameter	Value	Parameter	Value
-attention	global[7]	-learning_rate[8]	1
-beam_size	5	-learning_rate_decay	0.7
-brnn_merge	sum	-max_sent_length	250
-dropout[9]	0.3	-optim	sgd[10]
-dropout_type	naive[11]	-rnn_size	750
-encoder_type	brnn	-rnn_type	LSTM
-end_epoch	20	-vocab_size	100,000
-layers	2	-word_vec_size	500

Table 9.4: Parameter settings for the OpenNMT LSTM engine.

The LSTM-based Pictograph-to-Text translation tool proposes a list of hypotheses by generating all possible natural language text variants for each pictograph string. As several hundreds of thousands of hypotheses could be generated this way, we prune the hypothesis list at 10,000 sentences (which correspond to the 10,000 best sequences according to the n-gram language model). The LSTM re-scores the hypothesis list, outputting the winning sentence for each document.

9.2.3 Evaluation of the Baseline System

We present the results for Sclera-to-Dutch and Beta-to-Dutch translation. We use the test set (see Appendix B.2) of 50 Dutch messages and manually translate them into pictographs. Note that the source sentences in this task correspond to the reference translations that were used for evaluating the Text-to-Pictograph translation system. We did not use authentic pictograph messages, because we cannot always be sure of their intended meaning. Therefore, for this task, we are evaluating the scenario of back-translation into natural language text.

Note that, as we did not lowercase our data while training the 5-gram language

[7] Global attention attends to all source states/words (whereas local attention attends a subset of source states/words at a time).

[8] The pace/speed with which the neural network learns. A network is usually trained starting from a low learning rate, which is increased exponentially for every batch.

[9] Dropout addresses the problem of overfitting. The idea is to randomly drop units (along with their connections) from the neural network during training. This prevents them from co-adapting too much (Srivastava et al. 2014).

[10] Stochastic gradient descent.

[11] The dropout for each time step is different.

models, the language modelling-based Pictograph-to-Text translation systems can generate uppercased or truecased tokens, where deemed appropriate. However, for the purpose of this evaluation, the source sentences, reference translations, and system output are all lowercased.

Condition	BLEU↑	NIST↑	WER↓	PER↓	MET.↑	TER↓
Sclera						
5-gram mixed-dom. corpus	05.44	3.00	63.28	58.62	33.60	65.95
5-gram CGN corpus	06.30	2.67	63.96	60.27	32.99	66.46
5-gram SoNaR subt. corpus	**08.26**	**3.01**	**62.40**	**57.66**	**35.21**	**65.03**
LSTM mixed-dom. corpus BRNN 750	03.52	2.47	78.78	70.83	22.02	81.74
LSTM CGN corpus BRNN 750	04.05	2.85	73.55	65.50	26.34	75.69
LSTM SoNaR subt. corpus BRNN 750	03.75	2.04	87.80	81.98	18.13	88.51
Beta						
5-gram mixed-dom. corpus	07.59	3.24	62.69	58.72	41.72	82.60
5-gram CGN corpus	09.42	**3.45**	**60.17**	**55.52**	**39.34**	**60.72**
5-gram SoNaR subt. corpus	**09.73**	3.43	61.14	56.78	38.46	62.26
LSTM mixed-dom. corpus BRNN	06.92	3.06	70.35	64.63	31.44	71.49
LSTM CGN corpus BRNN 750	07.47	3.34	65.89	59.69	33.45	67.38
LSTM SoNaR subt. corpus BRNN 750	07.14	3.31	67.25	61.05	33.13	68.41

Table 9.5: Evaluation of Pictograph-to-Dutch translation using language models.

Table 9.5 shows the respective **BLEU** (Papineni et al. 2002), **NIST** (Doddington 2002), **Word Error Rate** (WER), **Position-Independent Word Error Rate** (PER), **METEOR** (Denkowski & Lavie 2014), and **Translation Error Rate** (TER) scores for the translation of pictograph messages into Dutch. A description of BLEU, NIST, WER, and PER is given in section 3.3.1. METEOR, unlike BLEU, incorporates the use of stemming and synonyms by matching the surface forms of the words and backing off to stems and semantic classes (Koehn 2009:228).[12] TER measures the number of

[12]In METEOR 1.5, Dutch is only partially implemented. While the tool offers flexible word matching with “exact matches” and “stem matches” for Dutch, it does not yet support “synonym matches” and “paraphrase matches” like it does for other languages, such as English or French. We build a version of METEOR for Dutch that uses the Cornetto (Vossen et al. 2008) links between concepts to match lemmas when they belong to the same synset or when they are hyperonyms, hyponyms, meronyms, subevents, holonyms, or near-synonyms. We also add a bilingual phrase table derived from Moses phrase tables (Koehn et al. 2007) trained on Europarl (Koehn 2005) and Opus (Tiedemann 2009) to further improve the matches.

edits required to change a system output into one of the references.[13]

Using bootstrap resampling (Koehn 2004), we calculate significance for BLEU between different pairs of conditions. For Sclera, we are able to measure a significant improvement when using the 5-gram language model over the LSTM language model in the mixed-domain corpus and SoNaR subtitles corpus conditions (but not in the CGN corpus condition). Similarly, for Beta, significant improvement is measured when using the 5-gram language model over the LSTM language model in the CGN corpus and SoNaR subtitles conditions (but not in the mixed-domain corpus condition). This observation indicates that, in most cases, using 5-gram language models for finding the most likely combination of lexical surface forms is better than using neural approaches in the language modelling task.[14] Furthermore, generating the hypotheses for the LSTM and re-scoring them is slow, making this technique less useful for real-time applications.

We also calculate significance for BLEU between the different corpora in the 5-gram language modelling approaches. For both pictograph languages, we find a significant difference between the model that uses the SoNaR subtitles corpus and the model that uses the large-mixed domain corpus, indicating that using more text from many different sources (including, for instance, proceedings from the European Parliament) is not necessarily better than using less text that appropriately models the target users' language (i.e., many first and second person forms, references to daily life, etc.).[15]

[13] In Sevens et al. (2015b), we reported a BLEU score of 05.93 for Sclera-to-Text and 07.39 for Beta-to-Text translation using a trigram language model on a (smaller) mixed-domain corpus using the same test set. We lowercased and detokenised the output and obtained a BLUE score of 07.36 for Sclera-to-Text and 07.92 for Beta-to-Text translation. While this language model is not able to outperform the 5-gram SoNaR subtitle corpus language model, it should be noted that it does not perform significantly worse (or better, for that matter) than the 5-gram mixed-domain corpus language model. This suggests that higher order *n*-grams do not necessarily lead to better results, possibly because exact 5-gram or 4-gram matches are more difficult to find in the corpus.

[14] In future work, we will investigate whether combining the two models, i.e., by using the LSTM language model in the beam search decoder instead of the 5-gram language model, leads to improved results.

[15] A similar observation is made by Dusserre et al. (2017), who evaluate the contributions of small, domain-specific data on a Word2Vec task. Their results show that, even though the general tendency of the community is to focus on using an increasingly large amount of data, the specificity of the corpus has much more influence on Word2Vec results than its size.

9.2.4 Shortcomings of the Baseline System

The baseline Pictograph-to-Text translation engine focusses on ease of composition, without sacrificing the user's freedom of expression. It requires a minimal set of resources, and can easily be extended to other languages (see Chapter 10). Evaluations of the baseline system show that using a language model for finding the most likely combination of the pictographs' lexical surface forms already produces decent translations, but there is still ample room for improvement.

It is important to note that the baseline models assume that the grammatical structure of pictograph languages resembles and simplifies that of a particular natural language. Nevertheless, the users of pictograph languages do not always need to introduce pictographs in a canonical order, or they could omit some of them. And while articles are added to our hypotheses, no other function words are generated. Prepositions, for instance, will never show up in the pictograph output.

In the next sections, we seek alternative solutions to the natural language generation problem.

9.3 Purely Rule-Based Approaches

A purely rule-based approach toward Pictograph-to-Text translation is explored by Lemmens (2016), who designed a toy grammar for his Depicto translation engine for Sclera-to-Text translation. The main objective of Depicto is to make up for the lack of resources for language pairs involving pictograph languages in particular, and for the semantically underspecified nature of pictograph languages in general.

Depicto's structure is based on a cascade of two grammars that are written in the Head-Driven Phrase Structure Grammar formalism (Pollard & Sag 1994) and is developed and compiled in the Answer Constraint Engine (Packard 2015). The first grammar receives a series of lemmas as input, which are associated with the selected pictographs, and analyses them with the objective of obtaining a well-formed structure. If such a structure is found, the grammar creates a semantic representation of the pictograph sentence. The semantic representation may contain features which were absent in the underspecified pictograph input. For instance, articles can be inserted by means of a small set of phrase rules. The second grammar takes the semantic representation as input and generates hypothetically well-formed sentences.

Lemmens argues that Depicto's precision, i.e., its ability to produce well-formed output, and performance are high, but must concede that its coverage is very limited. Its lexicon covers only 30 of the (in total) more than 13,000 pictographs comprised by the Sclera set. The main reason for this limitation is that the lexicon, which has been handcrafted, is very costly to extend, especially since each new addition to the grammar of Sclera requires corresponding additions to the transfer grammar and to the target grammar. Furthermore, Depicto only accepts complete clauses or noun phrases as a start symbol for parsing.

As an assistive writing tool, Depicto needs improvement. Its analysis module imposes overly stringent constraints on the order of elements on the pictograph input. For instance, Depicto assumes that the pictographs adhere to subject-verb-object order, and that the input can always be resolved to a complete (simple) main clause or to a noun phrase. The analysis module fails to parse the input if these conditions are not met. Furthermore, as the translation engine does not filter or rank its output, we end up with multiple well-formed translations. Note that a most likely translation can, therefore, not be chosen unless statistical methods are incorporated.

Rule-based approaches are costly and they require resources that are developed in large part by hand. They put severe constraints on a user's freedom of expression by expecting syntactically and/or semantically well-formed input. Even though purely rule-based engines have their merits - like the fact that they are able to generate grammatical output -, we have chosen not to continue their development, and to focus on data-driven approaches instead.

9.4 Data-Driven Machine Translation-Based Approaches

Statistical machine translation systems implement a mathematical theory of probability distribution and probability estimation (Vandeghinste 2008:16). They learn a translation model from a bilingual parallel corpus, and a language model from a monolingual target corpus. At runtime, the best translation is searched for by maximising the probability according to the two models (Carl & Way 2003).

At first sight, the PRESEMT (Tambouratzis et al. 2017) approach seems appropriate, as a parallel corpus of "correctly" written Dutch text and pictograph sequences does not exist, and a small bilingual parallel corpus could be crafted by hand (although this step would still require us to manually translate several hundreds of sentences into pictographs). PRESEMT uses a small bilingual parallel corpus and a large target language monolingual corpus, which are collected as far as possible over the web, primarily to simplify the development of resources for new language pairs. The only tools required are a tagger and lemmatiser for both languages and a shallow parser for the target language. However, PRESEMT is not (yet) able to compete with established systems such as Moses (Koehn et al. 2007). Furthermore, it requires an existing bilingual dictionary between the source language and the target language, while Moses extracts this information automatically from the parallel corpus (Koehn et al. 2007).

We will not consider this approach, since we found that it is possible to construct a large parallel corpus automatically. This corpus can provide us, in its turn, with plenty of training data for established machine translation toolkits, making the PRESEMT approach largely superfluous. We present two methods for creating such a parallel corpus in section 9.4.1. We then present the **phrase-based machine translation** approach in section 9.4.2 and the **neural machine translation** approach in section 9.4.3. Finally, we evaluate both systems as compared to the language modelling baseline in section 9.4.4.

9.4.1 Constructing a Parallel Corpus

For the construction of a parallel corpus, we take the "subtitles" component of the SoNaR corpus (Oostdijk et al. 2013) (i.e., subtitles for all sorts of Dutch television shows, such as game shows, soap operas, and reality shows), as it is made up of a large amount of correctly written Dutch text (27.6M tokens), includes many references to daily life activities, objects, and emotions,[16] and contains a substantial amount of first and second person forms (as opposed to newspaper text), making it a suitable corpus to model the type of writings that are produced by the end users of the WAI-NOT environment.

Our first method uses Text-to-Pictograph translation. We automatically translate the monolingual corpus into Sclera and Beta pictographs using the Text-to-Pictograph translation engine. The baseline system is used (see Chapter 3), and the word sense disambiguation module is activated (see Chapter 6). We do not activate the spelling correction and simplification modules (see Chapters 4 and 5), as spelling correction is not necessary, and we allow pictograph input to be complex. We create a parallel corpus with pictograph sentences on the source side, and the original Dutch subtitles on the target side. In the following sections, we will refer to this corpus as the (Sclera

[16]This is especially the case in soap operas.

or Beta) **Text-to-Pictograph parallel corpus**. We also build a tuning corpus, using the same method, with 100K sentences (1.8M tokens) of the "books" component of the SoNaR corpus. Note that, using this method, any errors that can be attributed to the Text-to-Pictograph translation engine are propagated.

We also propose a different method. We lemmatise the corpus, and remove all words that are not nouns, verbs, or adjectives. A handful of adverbs, such as *morgen* 'tomorrow', and a number of interjections, such as *hallo* 'hello', are retained. The modified, lemmatised texts mimic pictograph input (where each pictograph is replaced by its most frequent lemma). The result is a parallel corpus with "broken" sentences on the source side, and the original Dutch subtitles on the target side. In the following sections, we will refer to this corpus as the **deconstructed parallel corpus**. We also build a tuning corpus, using the same method, with 100K sentences (1.8M tokens) of the "books" component of the SoNaR corpus.

9.4.2 Phrase-Based Machine Translation

Our phrase-based machine translation approach uses the Moses decoder (Koehn et al. 2007) in its phrase-based mode,[17] with *grow-diag-final-and* as its phrase alignment criterion,[18] and its re-ordering parameter set to *msd-bidirectional-fe*.[19] We use the large language model for Dutch (see section 9.2.2) to model the target language side, and we use a *distortion-limit*[20] of 0. The relevant parameter values that are used during training and decoding are shown in Table 9.6. Three translation models are learned: one from the Sclera Text-to-Pictograph parallel corpus, one from the Beta Text-to-Pictograph parallel corpus, and one from the deconstructed parallel corpus.

For our additional experiments with factored models (Koehn 2009), we add part-of-speech information as a feature to each token in the source and target language corpora of each parallel corpus. Factored models integrate additional linguistic markup at the word level. Each type of additional word-level information is called a factor. Ad-

[17] Phrase-based models use phrases as atomic units, which are estimated from parallel corpora.

[18] The *grow-diag-final-and* heuristic starts with the intersection of the two alignments and then adds additional alignment points.

[19] The re-ordering model learns different re-ordering behaviour for each phrase pair. *Msd-bidirectional-fe* uses the MSD model (three orientation types on the source and target phrases: Monotone, Swap, and Discontinuous), and calculates the re-ordering orientation for the previous and the next word, for each phrase pair.

[20] This parameter is the absolute of the difference of the last word of the previously translated phrase and the position of the first word in the currently translated phrase.

Parameter	Value
-alignment	grow-diag-final-and
-dl	0
-lm	5 (KENLM)
-max-phrase-length	7
-reordering	msd-bidirectional-fe

Table 9.6: Parameter settings for the Moses phrase-based machine translation engine.

ditional information such as part-of-speech tags may be helpful in making re-ordering or disambiguation decisions. Like phrase-based models, factored models are a combination of several components. Each component defines one or more feature functions that are combined in a log-linear model (see Equation 9.1, where Z is a normalisation constant, e is the translation, f is an input sentence, and h_i is a feature function).

$$p(e|f) = \frac{1}{Z} exp \sum_{i=1}^{n} \lambda_i h_i(e,f) \qquad (9.1)$$

9.4.3 Neural Machine Translation

Linear models, such as phrase-based machine translation approaches, do not allow to define more complex relationships between features. Neural network models promise better sharing of statistical evidence between similar words and inclusion of rich context (Koehn 2017).

For each corpus, we train a neural machine translation engine using the OpenNMT toolkit (Klein et al. 2017), using mostly standard settings, and 500 hidden nodes. For the first experiment, we use a standard decoder. We then repeat the experiment with the best scoring corpus using a bidirectional encoder, and an additional 250 hidden nodes. The relevant parameter values that are used during training and decoding are shown in Table 9.7.

9.4.4 Evaluation

We show the results for Sclera-to-Dutch and Beta-to-Dutch translation. We use the test set of 50 Dutch messages (see Appendix B.2) and manually translate them into pictographs (see section 9.2.3).

Parameter	**Value**	**Parameter**	**Value**
-attention	global	-learning_rate	1
-beam_size	5	-learning_rate_decay	0.7
-brnn_merge	sum	-max_sent_length	250
-dropout	0.3	-optim	sgd
-dropout_type	naive	-rnn_size	500 & 750
-encoder_type	brnn	-rnn_type	LSTM
-end_epoch	20	-vocab_size	50,000
-layers	2	-word_vec_size	500 & 750

Table 9.7: Parameter settings for the OpenNMT neural machine translation engine.

In Table 9.8, we present the **BLEU** (Papineni et al. 2002), **NIST** (Doddington 2002), **Word Error Rate** (WER), **Position-Independent Word Error Rate** (PER), **METEOR** (Denkowski & Lavie 2014), and **Translation Error Rate** (TER) scores for the translation of pictograph messages into Dutch. For the purpose of this evaluation, the source sentences, reference translation, and system output are all lowercased.

Using bootstrap resampling (Koehn 2004), we calculate significance for BLEU between different pairs of conditions. When looking at the different types of models (*deconstructed* versus *Text-to-Pictograph*, and *factored* versus *non-factored*), we do not find a significant difference between any of the conditions, not for the Moses system, nor for the OpenNMT system.

When comparing the BLEU score for best-scoring version of the Moses system, i.e., the one that is trained on the factored Text-to-Pictograph parallel corpus, with the BLEU score for the best-scoring version of the OpenNMT system, i.e., the one that uses the bidirectional encoder with 750 hidden nodes, we find a significant difference for both pictograph languages. This indicates that the neural machine translation approach outperforms the phrase-based machine translation approach in the Pictograph-to-Text translation task. Significant improvement is also reported for both pictograph languages when comparing the output of the OpenNMT system trained on the Text-to-Pictograph parallel corpus using standard settings and 500 hidden nodes, with the OpenNMT translation system that uses a bidirectional encoder and 750 hidden nodes.

As compared to the baseline 5-gram language modelling system that uses the SoNaR subtitles corpus (see section 9.2.3), significant improvement is measured when using the best-scoring OpenNMT translation system for the Pictograph-to-Text transla-

Condition	BLEU↑	NIST↑	WER↓	PER↓	MET.↑	TER↓
Sclera						
Moses Deconstructed corpus non-fac.	06.27	3.06	76.65	64.53	27.21	75.49
Moses Deconstructed corpus fac.	05.86	**3.16**	73.06	62.31	27.92	72.31
Moses Text2Picto corpus non-fac.	05.58	2.75	67.83	59.88	30.42	68.62
Moses Text2Picto fac.	07.31	2.92	**63.37**	**57.27**	**33.71**	**65.44**
OpenNMT Deconstructed corpus 500	07.52	2.86	78.20	67.25	25.87	76.31
OpenNMT Text2Picto corpus 500	07.88	2.61	76.84	67.64	28.18	69.74
OpenNMT Text2Picto corpus BRNN 500	08.51	2.84	75.10	65.02	29.55	68.41
OpenNMT Text2Picto corpus BRNN 750	**11.14**	2.90	75.29	66.18	30.87	68.00
Beta						
Moses Deconstructed corpus non-fac.	07.25	3.46	71.51	60.37	30.29	71.28
Moses Deconstructed corpus fac.	07.34	**3.58**	68.41	56.49	31.98	68.10
Moses Text2Picto corpus non-fac.	08.30	3.28	65.41	54.07	31.25	64.51
Moses Text2Picto fac.	09.34	3.44	**63.76**	**52.81**	**32.32**	**63.59**
OpenNMT Deconstructed corpus 500	07.85	3.07	75.10	64.24	28.03	77.94
OpenNMT Text2Picto corpus 500	08.15	2.96	70.64	57.85	28.84	67.59
OpenNMT Text2Picto corpus BRNN 500	09.01	2.92	67.05	54.75	28.63	68.31
OpenNMT Text2Picto corpus BRNN 750	**12.93**	3.33	66.57	55.23	30.76	65.74

Table 9.8: Evaluation of Pictograph-to-Dutch translation using machine translation.

tion task. For the other machine translation models, including the OpenNMT systems that use different settings, we are not able to measure such significant improvements over the baseline systems.

Note that not all evaluation metrics point toward the same direction. For instance, WER, PER, METEOR, and TER favour the Moses-trained factored Text-to-Pictograph parallel corpus, whereas NIST favours the Moses-trained factored deconstructed parallel corpus. The difficulty of using automated metrics lies in the fact that there may be many alternative correct translations for a single source segment (Rowda & Pospelova 2016). This is especially the case in the Pictograph-to-Text translation task. The system must translate from a poor, underspecified pictograph language (with one pictograph corresponding to multiple words and word forms) into a rich natural language. This also explains why the scores are low (and the error rates are high) in most conditions. For these reasons, we are including human judgments on adequacy, fluency, and ranking as an additional measure.

9.5 Adequacy, Fluency, and Pairwise Rankings

Adequacy indicates how much of the meaning expressed in the gold standard translation is also expressed in the target translation. It is measured by means of a 5-point scale:

- 5: All meaning is preserved
- 4: Most meaning is preserved
- 3: Some meaning is preserved
- 2: Little meaning is preserved
- 1: No meaning is preserved

Fluency, on the other hand, only looks at the target translation. Some of the criteria related to fluency are grammar, spelling, choice of words, and style. Fluency is also measured by means of a 5-point scale:

- 5: The language in the output is flawless
- 4: The language in the output is good
- 3: The language in the output is non-native
- 2: The language in the output is disfluent
- 1: The language in the output is incomprehensible

For each of the approaches presented above, we present adequacy and fluency ratings for its highest-scoring translation engine according to BLEU. These are: (a) the 5-gram language modelling approach that uses the SoNaR subtitles corpus, (b) the LSTM language modelling approach that uses the CGN corpus, (c) the phrase-based machine translation approach that uses the factored Text-to-Pictograph parallel corpus, and (d) the neural machine translation approach that uses the Text-to-Pictograph parallel corpus, with bidirectional encoding, and 750 hidden nodes. During scoring, we present the output sentences to be rated in a random way (and without revealing the system that produced the output), in order to reduce bias. Evaluations are carried out by one judge (i.e., the author). The results are presented in Table 9.9.

For both the Sclera and Beta condition, these results reveal that the machine translation-based approaches outperform the language modelling approaches in terms

Condition	Adequacy (median)	Fluency (median)
Sclera		
5-gram SoNaR subt. corpus	4	3
LSTM CGN corpus BRNN 750	3.5	2.5
Moses Text2Picto fac.	5	4
OpenNMT Text2Picto corpus BRNN 750	5	4.5
Beta		
5-gram SoNaR subt. corpus	4	3
LSTM CGN corpus BRNN 750	3	2
Moses Text2Picto fac.	5	4
OpenNMT Text2Picto corpus BRNN 750	5	4

Table 9.9: Adequacy and fluency ratings for four different versions of the Pictograph-to-Text translation engine.

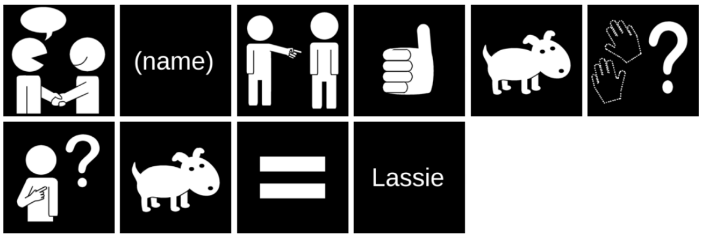
of meaning preservation and fluency. The 5-gram language modelling approach generates more adequate and fluent output than the LSTM approach, while comparable scores are attributed to the phrase-based machine translation and the neural machine translation output. We show three examples of Pictograph-to-Text translation using different versions of the translation engine in Table 9.15.

In the **pairwise ranking** task, judges are presented with multiple translations, and are required to choose the best option. The advantage of using ranking over rating information (adequacy, fluency) is that it is often easier for human judges to rank systems than to assign absolute scores (Vilar et al. 2007).

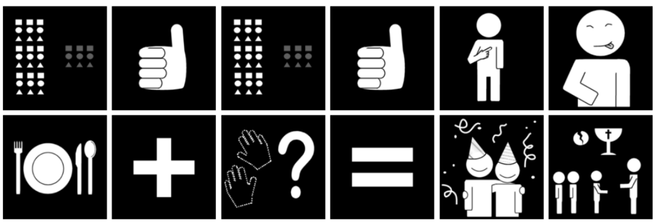
We apply a tournament strategy to further cut down the number of matches which need to be ranked and to circumvent the need for their explicit ordering. The implementation of the tournament strategy, developed for the SCATE project,[21] is based on the ranking approach proposed by Pighin et al. (2012). This approach breaks down a full-ranking task into pairwise comparisons, from which a global ranking of matches can be derived later on. Each annotator is presented one source sentence and a pair of alternative translations at a time. For each triplet, the annotators take a ternary decision, by marking the two translations as equivalent or by expressing a preference for one over the other. This approach is found to be faster, less laborious, and more consistent in terms of achieving higher interrater agreement than approaches using explicit many-to-many comparisons (Green et al. 2013).

[21] https://www.arts.kuleuven.be/ling/ccl/projects/scate

5-gram LM	goedemorgen *name* zijn je ouders komen?
LSTM LM	de hoi de *name* zijne jouw ouders kwamen?
Moses MT	hoi *name* zullen jouw ouders komen?
Neural MT	goedenavond *name* zijn uw ouders aangekomen?



5-gram LM	goedemorgen *name* je goed honden hoe heet honden zijn lassie
LSTM LM	goeiemorgen *name* je de goederen honden hoe genoemd honden zijn lassie
Moses MT	hoi *name* je favoriete hond hoe heet het hondje is lassie
Neural MT	goedenavond *name* je hebt een toffe hond hoe heet de hond is dat een lassie.



5-gram LM	heel goed heel goed ik lekker gegeten en hoe zijn vier communie
LSTM LM	meer best massa goeien mij lekker gegeten en hoe zij vierden de communie
Moses MT	heel makkelijk heel tof ik lekker eten en hoe lang vieren de communie.
Neural MT	heel redelijk heel tof ik heb lekker gegeten en hoe is de gevierde communie.

Table 9.15: Examples of Pictograph-to-Text translation using different versions of the translation engine.

We present system output (i.e., Pictograph-to-Text translations) for all 84 pictograph sequences (for each pictograph language) to four annotators. The annotators are native Dutch speakers who are familiar with the Sclera and Beta set. The pictograph sequences and system output are presented to the annotators in a random way, and information about the type of translation engine that was used to obtain each translation is removed, in order to reduce bias.

Agreement between annotators is calculated using Fleiss' kappa.[22] Cohen's kappa is calculated between each two annotators.[23] With respect to Fleiss' kappa, the agreement is 0.626 for Sclera and 0.629 for Beta. Both scores fall into the category of substantial agreement (0.61-0.80) according to the kappa interpretation scale (Landis & Koch 1977). Average pairwise Cohen's kappa coefficients between each two annotators range from moderate agreement (0.41-0.60) to almost perfect agreement (0.81-1.0) according to the kappa interpretation scale (see Table 9.16).

Sclera		**Beta**	
Annotators	**Cohen's kappa**	**Annotators**	**Cohen's kappa**
1 & 2	0.705	1 & 2	0.668
1 & 3	0.875	1 & 3	0.846
1 & 4	0.646	1 & 4	0.729
2 & 3	0.612	2 & 3	0.522
2 & 4	0.428	2 & 4	0.446
3 & 4	0.543	3 & 4	0.595

Table 9.16: Average pairwise Cohen's kappa coefficients between each two annotators.

The scores produced by the automatic metrics in the previous sections are correlated to the human evaluation produced by the "the median evaluator" [24] derived from the survey results using the Pearson correlation coefficient.[25] The ranking of matches produced by the metrics is also correlated to the normalised human ranking

[22]Fleiss' kappa is a way to measure agreement between three or more raters.

[23]Cohen's kappa is a measure of the degree of consistency between two raters.

[24]We produced the answers of the median evaluator by taking the median of the answers provided by the four evaluators for each pictograph sentence.

[25]The Pearson correlation coefficient is a number between -1 and 1 that indicates the extent to which two variables are linearly related.

using Spearman's rank correlation coefficient.[26]

We present the results for all 336 human evaluation/automated evaluation pairs (i.e., 84 pictograph sequences times 4 translation systems) per pictograph language in Table 9.17. Overall, the correlations are very low (-0.25 to 0.25) or low (-0.5 to -0.25 or 0.25 to 0.5). The metric which appears to correlate best with the human judgment of usefulness according to both rank and score is PER for the error rate-based metrics,[27] and METEOR for the other metrics. Note that the correlations between the human evaluations and BLEU are not statistically significant in the Sclera condition.[28]

	Sclera		**Beta**	
Metric	**Pearson**	**Spearman**	**Pearson**	**Spearman**
BLEU	0.0867	0.0686	0.1269*	0.1240*
NIST	0.1345*	0.1376*	0.2034**	0.2031**
WER	-0.2565**	-0.1980**	-0.2343**	-0.2426**
PER	-0.2622**	-0.2171**	-0.2779**	-0.2616**
METEOR	0.2773**	0.2696**	0.2428**	0.2150**
TER	-0.2578**	-0.2048**	-0.2567**	-0.2549**

Table 9.17: Correlation between human evaluation and all tested metrics based on the score (Pearson correlation coefficient) and rank (Spearman's rank correlation coefficient). $*p < 0.05, **p < 0.01$.

Figure 9.3 shows the rank distribution of the normalised human ranking per translation system for all 84 test sentences according to the median evaluator. For instance, for Beta, the output that is produced by the LSTM language modelling-based system is deemed the best translation (or a translation that is deemed equally as good as other translations) in 22 out of 84 cases, whereas the output that is produced by the neural machine translation system is deemed the best translation (or a translation that is deemed equally as good as other translations) in 60 out of 84 cases.

[26] Spearman's rank correlation coefficient measures the strength and direction of association between two ranked variables.

[27] Note that the correlation is negative, since there is an inverse relationship between the variables; lower error rates correspond to better systems.

[28] Previous studies on the schism between BLEU and manual evaluation have highlighted the poor correlation between machine translation systems and manual evaluation scores (Tan et al. 2015:74). Callison-Burch et al. (2006) present possible failures of BLEU by showing examples of translations with the same BLEU score, but of different translation quality.

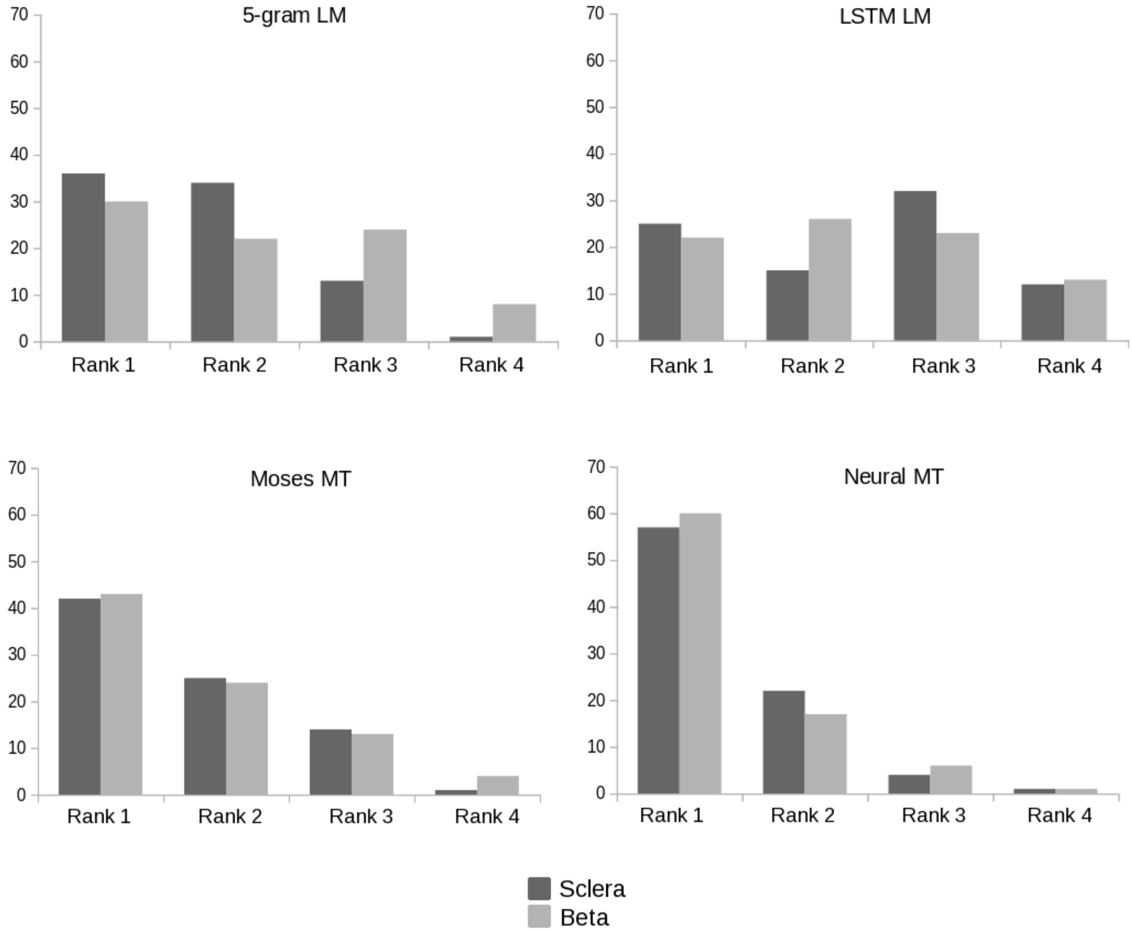


Figure 9.3: Rank distribution of the normalised human ranking per translation system for a total of 84 test sentences, as compared to the other three translation systems.

These graphs show that, in the majority of cases, human annotators prefer data-driven machine translation output over language modelling-based output. More specifically, for neural machine translation, there is a peak in rank number one, whereas the results for phrase-based machine translation are slightly more spread out over the different ranks. In other words, in most cases, the output produced by the neural machine translation is considered the best translation, or a translation that is deemed equally as good as other translations.

9.6 Conclusion: Development of a Pictograph-to-Text Translation Tool

In this chapter, we compared a number of approaches toward the generation of natural language text from pictograph input. Using automated metrics, we were able to demonstrate the added value of data-driven machine translation approaches over regular language modelling-based approaches. More specifically, significant improvements in BLEU were measured when using the neural machine translation system. Based on adequacy and fluency ratings, and pairwise rankings, we were able to confirm that neural machine translation is currently the most successful approach toward Pictograph-to-Text translation. In future work, we will experiment with different parameter settings in the OpenNMT environment, and add factored information to the neural machine translation pipeline.

Persona: Chara (After)

Thanks to the Pictograph-to-Text translation technology, Chara is now able to write her own status updates and messages using a combination of (automatically corrected) text and a very large set of Beta pictographs - offering her even more objects, actions, and feelings to choose from than her own pictograph cards ever did. She has already familiarised herself with the interface and finds it rather easy to retrieve her favourite pictographs. The translation engine automatically converts her messages into natural language text, which she can now post as a status update or comment, or send to her family and friends. Being able to communicate in online environments makes Chara feel more included.

Part IV

Extension and Comparison

CHAPTER **10**

Extending the Translation Technologies to Other Languages

The pictograph translation technologies described in the previous chapters were designed to be as language-independent as possible. Since our methods toward Dutch-to-Sclera/Beta and Sclera/Beta-to-Dutch translation can largely be transferred to other pictograph languages and natural languages - provided that a number of language-specific tools and resources are available -, new pictograph translation technologies do not have to be developed from scratch. In this respect, our technologies differ from other pictograph-based communication tools, such as the Prothèse Vocale Intelligente system (Vaillant 1998) or the Widgit Symbols[1] system (see Chapter 11), which largely focus on only one pictograph language or natural language at a time. Extending these systems to other languages would require developers to manually establish new connections between pictographs and words in those languages. Needless to say, this is a time-consuming task.

In this chapter, we demonstrate how a Text-to-Pictograph and Pictograph-to-Text translation tool can be built for pictograph languages other than Sclera and Beta, and natural languages other than Dutch. In section 10.1, we briefly explain the process of linking new pictograph sets to synsets in WordNets. Section 10.2 describes the development and evaluation of a baseline translation technology for English and Spanish, and proposes a number of methods toward enhanced syntactic and semantic analysis and improved natural language generation. We conclude in section 10.3.

[1] http://www.widgit.com/

10.1 Extending the Technologies to Other Pictograph Languages

In the previous chapters, we described a translation technology that uses Sclera and Beta pictographs (see section 1.3.2.3) as its target languages (in Text-to-Pictograph translation) or as its source languages (in Pictograph-to-Text translation).

Building pictograph translation technologies for pictograph sets other than Sclera and Beta requires an annotator to link the pictographs to synsets in a WordNet. This process can be sped up by means of a set of annotation tools developed by Vandeghinste & Schuurman (2014), which check whether there is an exact match between the pictograph's file name (without its file extension) and an entry in the WordNet database (for instance, *hond.png* and *hond* 'dog'). If this is the case, the annotator can select the appropriate sense of the entry, using the hyperonymy, synonymy, and antonymy information that is displayed for each entry to discriminate between senses. If this is not the case, the annotator can enter a synonym and then select the appropriate sense, or tell the annotation tool to connect the pictograph to multiple synsets, in the case of complex pictographs.

10.2 Extending the Technologies to Other Natural Languages: The English and Spanish Case

Within the framework of the EU-funded project Able-to-Include (see section 2.2.4.1), which aims to improve the living conditions of people with ID, we build English and Spanish versions of the pictograph translation systems.[2]

In section 10.2.1, we discuss how the existing links between WordNets can be used to automatically connect pictographs to words in source languages other than Dutch. Sections 10.2.2 and 10.2.3 describe how the language-independent architecture of the baseline systems can be used to build the English and Spanish versions of the Text-to-Pictograph translation tool and the Pictograph-to-Text translation tool, respectively.

The aim of this case study is to show that a basic pictograph translation engine can be built for any language for which a number of tools and resources are available.

[2]Previous versions of sections 10.2.1 to 10.2.2 appeared in Sevens et al. (2014) and Sevens et al. (2015a).

While we certainly do not exclude the possibility of adapting the more advanced solutions that we developed for Dutch (deep linguistic analysis for pictograph simplification, context-sensitive spelling correction, neural Pictograph-to-Text translation, etc.) toward English or Spanish, the practical implementation of these methods is beyond the scope of this work. However, we will provide a number of tools and guidelines for the interested reader to get started on the development of advanced technological solutions for natural languages other than Dutch.

10.2.1 Automatically Linking Pictographs to Other WordNets

An essential step in building pictograph translation systems for other languages is ensuring that the pictographs are connected to (sets of) words in those languages. Manually linking thousands of pictographs to synsets is a time-consuming procedure. Instead, by transferring the connections automatically, this process can be sped up drastically (see Figure 10.1).

The connections between WordNets are an important resource in knowledge-based multilingual language processing. The Cornetto database for Dutch, used to build the Dutch pictograph translation systems, contains connections to the English Princeton WordNet. In section 10.2.1.1, we describe how we automatically connect Beta and Sclera pictographs to synsets in Princeton WordNet 3.0.

Nowadays, most WordNets contain high-quality links between the synsets in the source language and Princeton WordNet 3.0, which is often viewed as the “central” WordNet. Having obtained the links between Beta and Sclera pictographs and Princeton WordNet 3.0, it becomes possible to automatically assign pictographs to synsets in any WordNet that is linked to Princeton WordNet.[3] For example, with the English pictograph connections in place, a mapping between the pictographs and Spanish synsets in the Multilingual Central Repository (MCR) 3.0 becomes possible. This process is described in section 10.2.1.2.

10.2.1.1 Connecting Pictographs to Princeton WordNet 3.0

The equivalence relations in Cornetto establish connections between Dutch and English synsets in Princeton WordNet version 1.5 and 2.0. We update these links to Princeton WordNet version 3.0 by using the mappings that are made available by

[3] A full list of linked WordNets can be found at http://globalwordnet.org/wordnets-in-the-world/

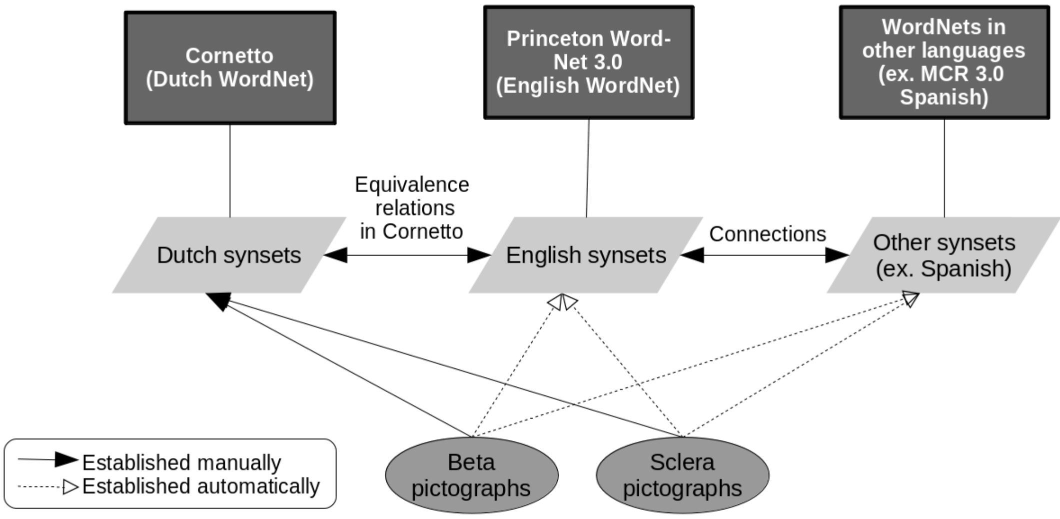


Figure 10.1: Transferring the pictograph links to Princeton WordNet 3.0 and WordNets in other languages, such as the MCR 3.0 for Spanish.

TALP-UPC.[4]

The equivalence relations between Cornetto and Princeton were established semi-automatically by Vossen et al. (1999), who carried out manual coding for the 14,749 most important concepts in the database, i.e., frequent concepts, concepts that have a large amount of semantic relations, and concepts that occupy a high position in the lexical hierarchy (i.e., more general concepts). All other concepts were linked automatically by mapping the bilingual Van Dale database[5] to Princeton WordNet 1.5: Once a matching synset had been found, all senses of that synset were proposed as possible translations. If there was only one translation, the procedure stopped, and the translation was assumed correct. If there were multiple translations, they were weighted using several heuristics, for instance, by measuring the conceptual distance in the WordNet hierarchy.

10.2.1.1.1 Improving the Equivalence Relations between Cornetto and Princeton WordNet We manually evaluated the quality of the links between 300 randomly selected Cornetto synsets and their (supposedly) equivalent Princeton synsets. Note

[4]http://www.talp.upc.edu/content/wordnet-mappings-automatically-generated-mappings-among-wordnet-versions

[5]http://www.vandale.be

that a Cornetto synset is often linked to more than one Princeton synset.

A Cornetto synset has an average of 3.3 automatically derived English equivalents. This observation allows us to compare our results to the initial quality check of the equivalence relations performed by Vossen et al. (1999), who note that, in the case of synsets having three to nine equivalents, the percentages of correct automatically derived equivalents go down to 65% and 49% for nouns and verbs, respectively. Our manual evaluations are in line with these results, showing that only 64.73% of all connections in our sample of 300 randomly selected synsets are correct; we found an erroneous link in no less than 35.27% of the 998 equivalence relations.

An example of where it goes wrong is the Cornetto synset for the animal *tor* 'beetle', which is not only appropriately linked to its correct synset (i.e., the Princeton synset containing the lemmas *beetle* and *bug*), but also mistakenly linked to the Princeton synset for the computational *glitch*. This flaw is most likely caused by its synonym *bug*, which is a commonly used term to denote errors in computer software. Examples like these are omnipresent in our data[6] and lead us to conclude that the synset links between Cornetto and Princeton WordNet need to be improved.

Translation dictionary	**Reference**	**Compilation**	**Number of word pairs**
Wiktionary	www.wiktionary.org	Manual	23,575
FreeDict	www.freedict.com	Manual	49,493
Europarl	Koehn (2005)	Automatic	2,970,501
Opus	Tiedemann (2009)	Automatic	6,223,539
Sclera file names	www.pictoselector.eu	Manual	12,381
Total			9,279,489

Table 10.1: The used translation resources.

We build a bilingual dictionary for Dutch and English and use these translations as an automatic indicator of the quality of the equivalence relations. We merge several translation word lists, and we remove double entries. Some resources are manually compiled dictionaries, while others are automatically derived word lists from parallel corpora. For instance, we extract the 1-word phrases from the phrase tables built with Moses (Koehn et al. 2007) based on the GIZA++ word alignments (Och & Ney 2003). Table 10.1 gives an overview.

[6]Other examples are *nederig* 'humble', which is linked to the synset for *flexible* (as a synonym for *elastic*), *waterachtig* 'aquatic', which is linked to the synsets for *grey* and *mousy*, and *rocker* 'hardrocker', which is linked to the synset for *rocking chair*.

For 52.18% (43,970 out of 84,264) of the equivalence relations, translation information is available in the bilingual dictionary in order to possibly confirm (or reject) the relation.

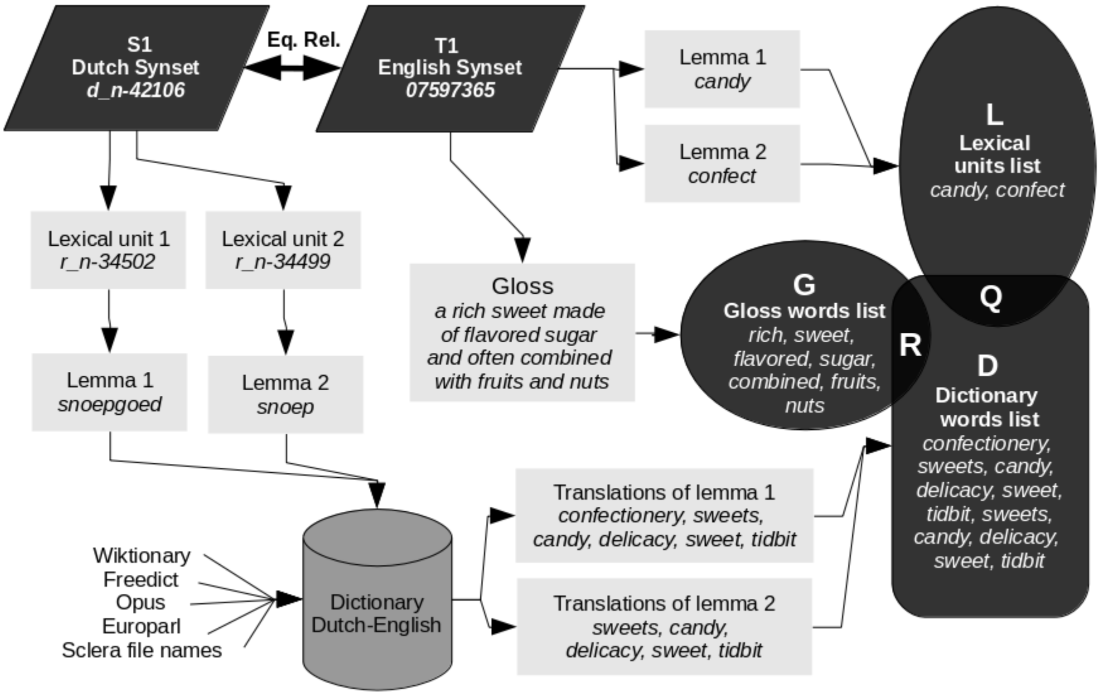


Figure 10.2: The scoring mechanism with examples.

Figure 10.2 visualises how we use the bilingual dictionary to automatically evaluate the quality of the pre-established links between Cornetto and Princeton WordNet.

We retrieve all the lemmas of the lexical units that are contained within a Dutch synset S_i (in this example, *snoepgoed* and *snoep*, extracted from S_1). We look up the lemmas in the bilingual dictionary and obtain a *dictionary words list D* of English translations.[7] This list is used to estimate the correctness of the equivalence relation between the Cornetto and the Princeton synset.

We retrieve the *lexical units list L* from the English synset T_j (in this example, *candy* and *confect*, extracted from T_1). We count the number of words in the lexical units list L that also appear in the dictionary words list D. In Figure 10.2, this overlap is represented as the multiset Q. Translations appearing more than once in the dictio-

[7]Note that this list may contain doubles (such as *candy* and *delicacy*), as these translations may provide additional evidence to our scoring algorithm. It is, therefore, not the case that the dictionary words list represents a *set*. It represents a *multiset*.

nary words list D are given more weight. For example, *candy* occurs twice. This puts the overlap counter on 2. The overlap is normalised: It is divided by 3 ($|Q|+|L\backslash Q|$, i.e., *confect* + *candy* + *candy*), leaving us with a score of 66.67%.

For the *gloss words list G*, we remove the stop words[8] and make an analogous calculation.[9] In this example, *sweet* occurs twice in the dictionary words list D. In Figure 10.2, this overlap is represented as the multiset R. This number is divided by 8 ($|R|+|R\backslash G|$, i.e., *sweet* + *sweet* + *rich* + *flavored* + *sugar* + *combined* + *fruits* + *nuts*). Averaging this score of 25% with our first result, we obtain a confidence score of 45.83% for this equivalence relation.

The formula to obtain confidence scores for the equivalence relations is shown in Equation 10.1.

$$Score = \frac{\frac{|Q|}{|Q|+|L\backslash Q|} + \frac{|R|}{|R|+|G\backslash R|}}{2} \tag{10.1}$$

We automatically calculate this confidence score for every equivalence relation in Cornetto.

We now verify whether the automatic scoring algorithm (dis)agrees with the manual judgments, in order to determine a satisfactory threshold value for the acceptance of equivalence relations. Evaluation results are shown in Figure 10.3. While the precision (i.e., the proportion of links that the system got right) goes slightly up as our criterion for link acceptance becomes stricter, the recall (i.e., the proportion of correct links that the system retrieved) quickly makes a deep dive. The F-score reveals that the best trade-off is reached when synset links with a confidence score of 0% are rejected, and any links with a positive confidence score are retained. The results in Table 10.3 show that, this way, we are able to reduce the error rate to 21.09%.

10.2.1.1.2 Improving the Equivalence Relations in the Context of Text-to-Pictograph Translation We also perform manual evaluations for a randomly generated subset of synsets that were used by Vandeghinste & Schuurman (2014) for assigning Sclera and Beta pictographs to Cornetto. Table 10.2 presents the coverage of the bilingual dictionary for only those synsets that are connected to Sclera and Beta pictographs. It can be seen that the coverage is much higher than the coverage over all synsets - this can be explained by the fact that Sclera and Beta pictographs cover

[8]https://www.ranks.nl/stopwords

[9]The glosses are included because they often contain synonyms or semantically related words.

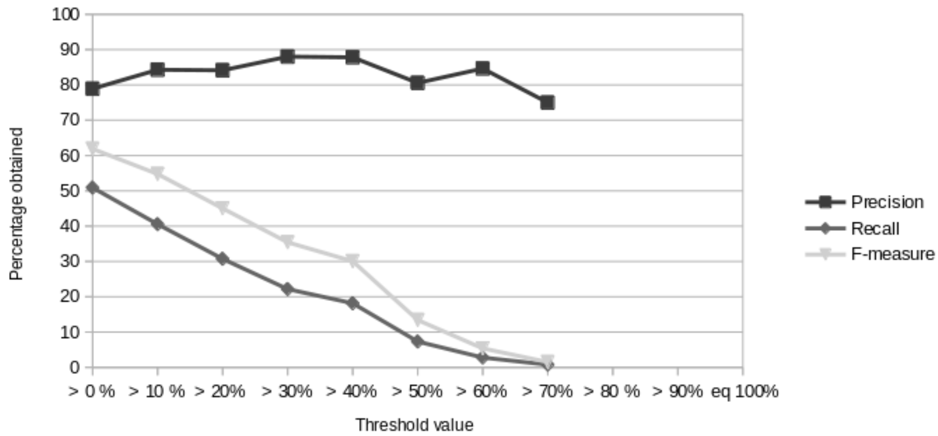


Figure 10.3: Precision, recall, and F-score for different threshold values of link acceptance.

everyday concepts, the lexical units of which are more common and therefore much easier to translate.

Condition	Covered	Total	Difference with all synsets
All synsets	43 970 (52.18%)	84 264	-
Sclera synsets	5 294 (88.80%)	5 962	36.62%
Beta synsets	3 409 (88.94%)	3 833	36.76%

Table 10.2: Dictionary Coverage for different sets of synsets.

Table 10.3 shows that the error rate of Cornetto's equivalence relations on the Sclera and Beta subsets is much lower than the error rate on the whole set. We attribute this difference to the fact that Vossen et al. (1999) carried out manual coding for the most important concepts in the database and that most Sclera and Beta pictographs also belong to this category.

In the Sclera and Beta cases, each synset has between one and two automatically derived English equivalents on average, allowing us to compare our results with the initial quality check of the equivalence relations performed by Vossen et al., who show that, in the event of a Dutch synset having only one English equivalent, 86% of the nouns and 78% of the verbs are correctly linked, while the ones having two equivalents are appropriate in 68% and 71% of the cases, respectively.

The F-score in Figure 10.4 reveals that the best trade-off between precision and

Condition	Baseline	Current	Relative improvement
All	35.27%	21.09%	40.20%
Sclera	14.50%	9.95%	31.38%
Beta	15.77%	13.47%	14.58%

Table 10.3: The reduction in error rates of Cornetto's equivalence relations.

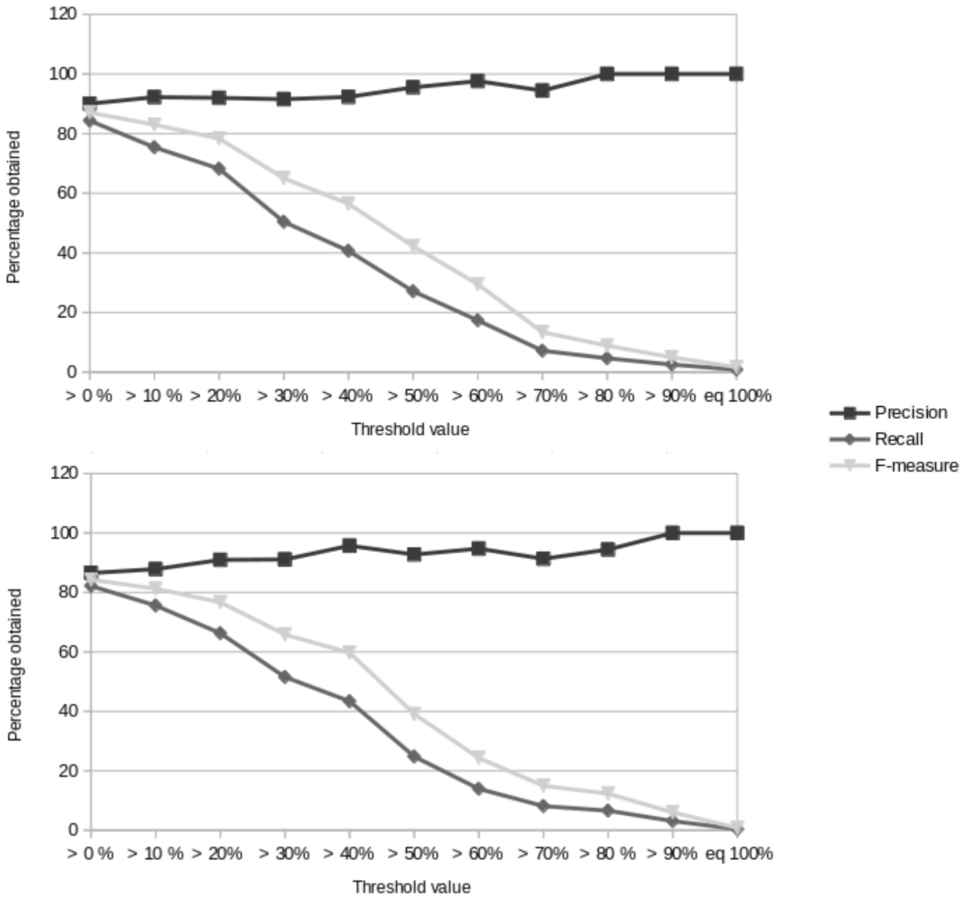


Figure 10.4: Precision, recall, and F-score for Sclera (top) and Beta (bottom) synsets, for different threshold values of link acceptance.

recall is again reached at the > 0% threshold value. In conclusion, we should retrieve all English synsets for which a non-zero score was obtained in order to automatically assign Sclera and Beta pictographs to Princeton WordNet.

As shown in Table 10.3, an error rate of 21.09% is now obtained for the general set of Cornetto's equivalence relations, while its subset of Sclera and Beta synsets (denoting frequent concepts) ends up with an error rate of 9.95% and 13.47%, respectively.

It now becomes possible to automatically assign a large amount of Sclera and Beta pictographs to English synsets in Princeton WordNet 3.0. Note that 154 (5.58%) Beta pictographs and 288 (5.04%) Sclera pictographs still have to be connected manually, either because the original equivalence relation was rejected by the filtering algorithm, or because a Dutch compound word corresponds to multiple words in English, which requires us to treat the simplex pictograph as a complex pictograph in English. One such example is the Dutch word *vanillesuiker*, which means *vanilla sugar* in English. In a handful of cases, an English variant cannot be created due to cultural differences. Think of the fictional character *Zwarte Piet* or typical kinds of food such as *choco*, which can roughly be translated as *chocolate spread*. We have decided not to include these pictographs in the English database. Still, by using the automated method, we were able to successfully avoid the manual linking of several thousands of pictographs.

10.2.1.2 Connecting Pictographs to the Spanish MCR 3.0

The MCR 3.0[10] integrates WordNets from five different languages, namely English, Catalan, Spanish, Basque, and Galician. Words in one language are connected to words in any of the other languages through inter-lingual-indexes. Using 300 randomly generated synsets, we verified these links between English and Spanish synsets and found that they were correctly established, making it possible for us to create highly reliable connections between Beta and Sclera pictographs and Spanish synsets. Observe that the exact same linking process can now be done for any language's WordNet that establishes reliable links to Princeton WordNet 3.0.

10.2.2 Text-to-Pictograph Translation for English and Spanish

In this section, we describe how English and Spanish written messages are converted into a sequence of Sclera or Beta pictographs. The translation process is essentially the same as for the baseline Dutch engine, although a number of language-specific resources must be provided. The focus in this section lies on the differences with the Dutch baseline system (see Chapter 3). We first describe the architecture of the

[10]http://adimen.si.ehu.es/web/mcr/

baseline translation systems in section 10.2.2.1. Section 10.2.2.2 presents a manual evaluation. While the development of advanced translation modules for English and Spanish is beyond the scope of this work, we propose a number of resources and methods toward building improved Text-to-Pictograph translation systems for languages other than Dutch in section 10.2.2.3. Finally, we conclude in section 10.2.2.4.

10.2.2.1 System Architecture

The source text first undergoes **shallow linguistic processing**, consisting of several sub-processes. This process is analogous to the linguistic processing step in the original Dutch tool, but uses some specific resources.

First, *tokenisation* is applied to split punctuation signs from words. Basic *spelling correction* (one deletion, one insertion, one substitution) aids in finding the correct variant of words that do not appear in the lexicon[11] and the list of first names.[12] Next, *part-of-speech tagging* is applied. For English, we use HunPos (Halácsy et al. 2007), an open-source tagger, using the English training data made available on the HunPos website.[13] For Spanish, part-of-speech tagging and lemmatisation are done in one step with TreeTagger (Schmid 1995).[14] TreeTagger is available for a large variety of European languages. The English tagger uses the Penn Treebank tagset,[15] whereas the Spanish tagger uses the TreeTagger tagset.[16] The next step is *lemmatisation*, which requires a language-specific treatment. For English, we build a lemmatiser based on the American National Corpus (Ide & Suderman 2004) lexicon,[17] as it contains English token/part-of-speech combinations and their lemma.[18] Finally, *sentence detection* is applied, using the same method for all three languages.

An additional adaptation concerns the treatment of the Spanish *pro-drop* phenomenon (which occurs in all Romance languages, with the exception of French), i.e., the omission of personal pronouns in subject position (unless emphasis is given).

[11] For English: http://www.anc.org/data/anc-second-release/frequency-data/; for Spanish: http://corpus.leeds.ac.uk/frqc/internet-es-forms.num

[12] http://www.quietaffiliate.com/free-first-name-and-last-name-databases-csv-and-sql (for English and Spanish)

[13] https://code.google.com/p/hunpos/downloads/list

[14] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

[15] https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

[16] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/data/spanish-tagset.txt

[17] See footnote 11.

[18] Note that, due to the data being readily available, we chose to use the American National Corpus for the shallow linguistic analysis step; we did not create a British variant.

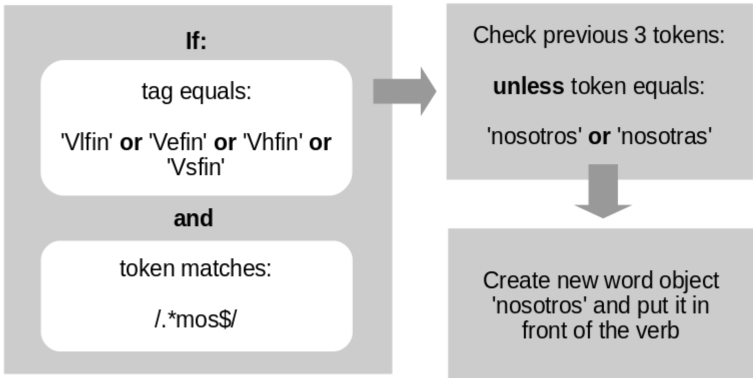


Figure 10.5: Example of a pro-drop rule. The tags correspond to *finite lexical verb*, *finite estar (to be)*, *finite haber (to have)*, and *finite ser (to be)*. The token should end on *-mos*, indicating a first person plural form. *Nosotros* and *nosotras* correspond to the English pronoun *we*.

When such a message is translated into pictographs, there would be no subject pictograph, as the pictographs are based on the lemma form of the tokens and do not use inflectional information. However, person information can be inferred from the verb in the source sentence. We write a set of rules that generate personal pronouns before converting the message into pictographs.[19] When a matching personal pronoun is already found within a window of three words, the rules are not applied (see Figure 10.5).[20]

The first step in the **semantic analysis** consists of detecting words with a negative polarity. For English and Spanish, these words are *not/no* and *no/ningún*.[21] For each word in the source text, the system returns all possible WordNet synsets. At this point,

[19]When the verb is a third person singular or plural, these rules are not applied, as its subject could as well be a noun phrase. This problem can be solved by applying deeper grammatical analysis. Gender information (*he*, *she*, *it*), however, cannot be inferred from the verb alone and requires deeper semantic knowledge.

[20]We evaluate the accuracy of the pro-drop resolution module on a test set of 50 Spanish tweets (see section 10.2.2.2). Out of the 31 occurrences of first or second person pro-drop, 25 cases (80,65%) are resolved correctly, and an appropriate pronoun pictograph is generated. We also observe 8 cases of third person pro-drop; as mentioned before, these cases are not treated by the pro-drop module.

[21]Note that this method is not robust. In the case of *not* (in English) and *no* (in Spanish), for instance, the system looks for a verb in the three preceding or three following words. As a result, a sentence pair such as *not all items are for sale* and *all items are not for sale* will receive the exact same semantic treatment, even though the two sentences have a different meaning. For Dutch Text-to-Pictograph translation, we were able to solve this issue by means of a full syntactic parse (see Chapter 5).

the language-specific WordNets are consulted. As shown in section 10.2.1, Princeton WordNet 3.0 and MCR 3.0 are used for English and Spanish, respectively. In order to remove unwanted meanings of common words, which are not appropriate for some groups of users (such as one meaning of the lemma *member*), we manually disable certain links between lemmas and synsets.

In section 10.2.1, we described how Princeton 3.0 and MCR 3.0 synsets were connected to pictographs. Just like Cornetto, the English and Spanish WordNets contain links between synsets, which can be used to look for alternative pictographs with a similar meaning if no pictograph for the proper concept is found. The use of WordNet relations is controlled by a number of parameters or penalties.

To make sure that *personal and possessive pronouns* are covered, they are given an explicit treatment. We manually store all English and Spanish pronouns, along with their part-of-speech tag, in a pronoun database. The **dictionary** is used to cover any words that are missing from the WordNet, either because their part-of-speech tag is not included in the WordNet database (such as various types of greetings, such as *hola* in Spanish), or because the concept is too recent (such as *tablet*), among other things.

For every word in the sentence, the system checks whether one or more pictographs can be found for it and whether the use of these pictographs is subject to a penalty. Penalties correspond to parameters that are tuned beforehand. The English and Spanish systems use the exact same types of parameters as the Dutch baseline system (see section 3.2.1).

The **parameters** have to be tuned for every natural language/pictograph language pair. Ideally, tuning is done using emails or text messages written by people with ID, such as the WAI-NOT corpus for Dutch (see section 2.2.4.2). As we do not have a large corpus of messages written by the target users at our disposal, we selected 50 English tweets[22] and 50 Spanish tweets based on the following criteria: the messages should contain at least eight words, they must refer to personal experiences (in other words, we do not include newspaper headlines or lyrics), and they can contain spelling mistakes or lack punctuation marks. The tweets are retrieved by searching for messages containing the hashtags *#school/#escuela*, *#love/#amor*, *#family/#familia*, *#happy/#feliz*, and *#sad/#triste*.

For both languages, we manually translate, to the best of our ability, all tweets into Beta and Sclera pictographs. We then use the local hill climber (Vandeghinste

[22]https://twitter.com/

				English		Spanish	
Parameter	Min	Max	Step	Sclera	Beta	Sclera	Beta
WordNet relations							
Threshold	5	20	1	7	19	10	10
Hyperonym penalty	0	15	1	8	7	3	4
XPos penalty	0	15	1	10	10	2	1
Antonym penalty	0	15	1	7	9	6	10
Pictograph features							
Wrong number	0	10	1	0	0	1	2
No number	0	10	1	2	4	4	5

Table 10.4: Parameter values for the English and Spanish Text-to-Pictograph translation systems after tuning.

et al. 2017) to vary the parameters when running the Text-to-Pictograph translation script on each of the four test sets (from English and Spanish to Beta and Sclera). A more thorough description of this algorithm is provided in section 3.2.2. We run five trials of the algorithm and take the best scoring parameter values for all four natural language/pictograph language pairs. We show the results in Table 10.4.

An A* algorithm calculates the **optimal pictograph sequence**. The optimal path selection process does not differ from the one that is used in our original Dutch translation system. For more details, we refer to section 3.1.

10.2.2.2 Evaluation

Since we do not have a corpus of messages written by people with ID at our disposal, an evaluation set is built using the selection procedure described in the previous section. We retrieved a total of 75 English tweets and 75 Spanish tweets. We use the same method for manual evaluation as described in section 3.3.2, in which we evaluate the baseline translation system for Dutch.

The results for English and Spanish are shown in Tables 10.5 and 10.6, respectively. Using the automatic pictograph connections that we created by using the links between Cornetto and Princeton WordNet and the links between Princeton WordNet and the Spanish MCR 3.0, a baseline system is built. This system leaves us with **F-scores** of 66.50% and 73.36% for Sclera and Beta, respectively, for English text without proper names. For Spanish, F-scores of 65.16% and 70.66% are obtained.

		Without proper names		With proper names	
Condition	**Precision**	**Recall**	**F-score**	**Recall**	**F-score**
Sclera					
Baseline	71.37%	62.25%	66.50%	61.25%	65.92%
Add frequent concepts	93.30%	73.04%	81.94%	71.95%	81.25%
Rel. improv.	*30.73%*	*17.33%*	*23.22%*	*17.47%*	*23.26%*
Beta					
Baseline	75.08%	71.71%	73.36%	70.63%	72.78%
Add frequent concepts	82.56%	86.14%	84.31%	85.07%	83.80%
Rel.improv.	*9.96%*	*20.12%*	*14.93%*	*20.45%*	*15.14%*

Table 10.5: Manual evaluation of the English system.

		Without proper names		With proper names	
Condition	**Precision**	**Recall**	**F-score**	**Recall**	**F-score**
Sclera					
Baseline	73.84%	58.30%	65.16%	57.63%	64.74%
Add frequent concepts	93.31%	83.14%	87.93%	82.17%	87.38%
Rel. improv.	*26.37%*	*42.61%*	*34.95%*	*42.58%*	*34.97%*
Beta					
Baseline	83.48%	61.26%	70.66%	60.83%	70.38%
Add frequent concepts	94.64%	86.83%	90.57%	86.01%	90.12%
Rel.improv.	*13.37%*	*41.74%*	*28.18%*	*41.39%*	*28.05%*

Table 10.6: Manual evaluation of the Spanish system.

To improve the English and Spanish systems, we manually check the 500 most frequently used words according to the Dutch WAI-NOT corpus. We translate each of these words into English and Spanish and check whether the correct pictograph is connected to their synset. If this is not the case, we disable the erroneous pictographs or we create new pictograph connections. The English system currently yields F-scores of 81.94% and 84.31% for Sclera and Beta, respectively, while the Spanish system reaches F-scores of 87.93% and 90.57%, for text in which proper names are omitted. These results are in line with the manual evaluations for the baseline translation system for Dutch. As shown in section 3.3.2, we obtained F-scores of 87.16% and 87.27% for Sclera and Beta translations of Dutch text, respectively.

10.2.2.3 Beyond the Baseline System

We created a baseline system for English and Spanish, which performs basic spelling correction, does not simplify the input text, and does not perform word sense disambiguation. In this section, we briefly describe how our modules for improved Dutch-to-pictograph translation could be adapted toward other languages.

Our approach toward automated **spelling correction for people with ID** does not require massive amounts of training data. To create the Dutch spelling correction system (see Chapter 4), we manually corrected 1,000 messages written by people with ID and used these corrections to infer character rewrite rules, using the language-independent Levenshtein Distance Alignment metric to align the sentences. To build English and Spanish versions of this module, we would need a comparable amount of messages that adequately models the type of language that is used by the target users. To our knowledge, there exist no websites like WAI-NOT for languages other than Dutch, so these user-generated messages would, ideally, have to be crowdsourced.[23] The module also uses a monolingual lexicon with frequencies, which checks whether a real word (with a minimum frequency) has been formed by the variant generation module. We already described a lexicon for basic English and Spanish spelling correction in section 10.2.2.1. Once all variants for a misspelled or infrequent real word have been generated, a trigram language model filters the total amount of possibilities. We present a language model for English and Spanish in section 10.2.3. Finally, the fuzzy matching step requires a monolingual corpus that adequately models the type of language of our end users, i.e., spontaneous interactions. For English, the orthographic transcriptions of unscripted informal conversations in the British National Corpus (BNC 2007)[24] could be used for this. One example of a Spanish corpus that could be useful is COLA,[25] the Spanish spoken corpus of youth language, which is transcribed orthographically. With all the resources in place, the parameters must be tuned for each language on a held-out set of user-generated messages.

As our module for **syntactic simplification** (see Chapter 5) is largely rule-based, transferring it to languages other than Dutch can be a challenging task. To create English and Spanish simplification modules, we must first study the characteristics of

[23] Another possibility would be to use Twitter data. However, as we showed for Dutch, tweets and text written by people with ID are different kinds of microtext. It is not safe to assume that using tweets as training data would have the same effects.

[24] http://www.natcorp.ox.ac.uk/

[25] http://www.colam.org/publikasjoner/COLA-cl2005-fig.htm

"difficult" syntactic phenomena in these languages, and determine which transformation rules could be applied to them in order to create active, independent, SVO-type clauses.[26] It is desirable to consult language-specific guidelines for easy-to-read language, such as the "Make it Simple" report on easy-to-read language by Freyhoff et al. (1998) for English, or the recommendations by Intertext[27] for Spanish. Building a language-specific simplification system also requires a syntactic parser for that language. For English, the Stanford Parser (Klein & Manning 2003) can be used. A Spanish pipeline is included in the Stanford Parser, as well.[28] Note, however, that adopting a rule-based approach for English and Spanish syntactic simplification might not be necessary, as parallel corpora of unsimplified and simplified text for these languages exist, thus paving the way for more advanced techniques, such as machine translation. Previous work on syntactic simplification for these languages has been done by Bott et al. (2012), Zhu et al. (2010), and Siddharthan (2014), among many others. The Simplext tool (Saggion et al. 2011),[29] which was developed within the framework of the Able to Include project (see section 2.2.4.1), is an open-source web service for English and Spanish that automatically simplifies input text as a pre-processing step for further natural language processing. For the advanced verb cluster simplification and temporality detection module, language-specific, hand-written rules must be provided.

The **word sense disambiguation** module for Dutch (see Chapter 6) uses an external tool that retrieves confidence scores for synsets. For the English Text-to-Pictograph translation engine, Jacobs (2015) implemented the word-sense disambiguation tool by Pedersen et al. (2005) into the translation pipeline, which computes the degree of string overlap between the context of a target word and the Princeton WordNet glosses for each sense that belongs to that word. We have no knowledge of tools that perform word sense disambiguation for Spanish using WordNet synsets, but Sobrevilla Cabezudo et al. (2015) propose a method that uses the PageRank algorithm to rank the senses associated to words to be disambiguated. With the word sense disambiguation tool in place, the translation system's parameters must be re-optimised, preferably using a tuning set of user-generated messages that contain plenty of polysemous words.

---

[26]It could be SOV for languages like Japanese, for instance.

[27]https://www.intertext.es/servicios/otros-servicios/adaptacion-textos-lenguaje-lectura-facil/

[28]https://nlp.stanford.edu/software/spanish-faq.html

[29]https://github.com/able-to-include/accessibility-layer/blob/master/Simplext.php

10.2.2.4 Conclusion: Text-to-Pictograph Translation for English and Spanish

We have described how the baseline Dutch-to-Pictograph translation system can be extended to other natural languages. To implement new languages, the following components are required: decent connections between the source language's WordNet and Princeton WordNet 3.0, a language-specific part-of-speech tagger and lemmatiser, a new set of parameters to optimise the system's performance, and (optionally) some rules to deal with language-specific properties.

10.2.3 Pictograph-to-Text Translation for English and Spanish

In this section, we describe how pictograph-based messages are translated into English and Spanish natural language text. Just like the Text-to-Pictograph translation engine, the Pictograph-to-Text translation process is essentially the same as for the baseline Dutch engine, with the exception of a number of language-specific resources. The focus in this section lies on the differences with the Dutch 5-gram baseline system (see section 9.2). We first describe the architecture of the baseline translation engines in section 10.2.3.1. An evaluation of the translation systems is provided in section 10.2.3.2. While the development of advanced machine translation modules for English and Spanish is beyond the scope of this work, we propose a number of resources and methods toward building improved Pictograph-to-Text translation systems for languages other than Dutch in section 10.2.3.3. Section 10.2.3.4 concludes.

We will not consider the development of a language-specific pictograph interface, as its design is largely language-independent and fully depends on concepts, rather than natural language text. Note that pictographs may be removed from the interface or replaced by pictographs that are more appropriate or recognisable for another culture, if deemed necessary. The pictograph prediction engine receives pictograph input and generates pictograph output, and can, therefore, also be re-used for other natural languages without further adaptations.

10.2.3.1 System Architecture

When a pictograph is selected by a user, its synset is retrieved, and from this synset, we retrieve all the synonyms it contains. For each of these synonyms, we apply reverse lemmatisation, and we retrieve the full, inflected linguistic paradigm of the lemma, together with its part-of-speech tag. For English, we build a reverse lem-

				English		Spanish	
Parameter	Min	Max	Step	Sclera	Beta	Sclera	Beta
Threshold pruning	0	20	1	17	15	2	1
Histogram pruning	0	20	1	4	9	2	8
Cost	0	20	1	6	12	15	9
Rev. lem. minimum frequency	2	41	1	36	11	36	11

Table 10.7: Parameter values for the English and Spanish Pictograph-to-Text translation systems after tuning.

matiser based on the American National Corpus.[30] The Spanish reverse lemmatiser combines FreeLing v.3.1's lemma list[31] with a Spanish word frequency list.[32] Each of the surface forms generated by the reverse lemmatiser is a hypothesis for the language model. For nouns, the system generates additional alternative hypotheses which include an article, based on part-of-speech information. This is especially important for Spanish, which is a morphologically rich language.

The training corpus that we use to build the English language model combines the Europarl corpus (Tiedemann 2012), the DGT corpus (Tiedemann 2012), and the British National Corpus.[33] The Spanish training corpus merges the Europarl corpus, Wikipedia entries,[34] and the Childes corpus.[35]

The system performs beam search decoding on a 5-gram language model with interpolated modified Kneser-Ney smoothing (Chen & Goodman 1999), trained with the KenLM Language Model Toolkit (Heafield et al. 2013) on the training corpora.

The 5-gram-based Pictograph-to-Text translation system contains a number of decoding parameters. The English and Spanish systems use the exact same types of parameters as the Dutch baseline system (see section 9.2.2.1). These parameters are tuned for every natural language/pictograph language pair using a local hill climber (Vandeghinste et al. 2017) on the parameter search space. The tuning set consists of 50 manually translated messages from Twitter. Note that the source sentences in this task correspond to the reference translations that we used for tuning the Text-to-

[30] See footnote 11. Note that, due to the data being readily available, we chose to use the American National Corpus for the shallow linguistic analysis step; we did not create a British variant.

[31] http://nlp.lsi.upc.edu/freeling/

[32] http://invokeit.wordpress.com/frequency-word-lists/

[33] http://www.natcorp.ox.ac.uk/

[34] https://www.wikipedia.org/

[35] https://childes.talkbank.org/

Pictograph translation systems (see section 10.2.2.1). The optimal parameter settings for English and Spanish are presented in Table 10.7.

10.2.3.2 Evaluation

We use our test set of 75 English and Spanish tweets and manually translate them into pictographs. Note that the source sentences in this task correspond to the reference translations that we used for evaluating the Text-to-Pictograph translation systems in section 10.2.2.2. For the purpose of this evaluation, the source sentences, reference translation, and system output are lowercased.

Condition	BLEU↑	NIST↑	WER↓	PER↓	MET.↑	TER↓
Sclera	07.32	2.80	71.22	68.13	19.97	71.99
Beta	08.66	3.21	66.98	64.66	21.69	67.42

Table 10.8: Evaluation of Pictograph-to-English translation using a 5-gram language model.

Condition	BLEU↑	NIST↑	WER↓	PER↓	MET.↑	TER↓
Sclera	07.18	1.97	73.17	72.02	25.45	74.84
Beta	07.47	2.31	71.53	70.79	30.75	72.07

Table 10.9: Evaluation of Pictograph-to-Spanish translation using a 5-gram language model.

Table 10.8 and Table 10.9 present the results for English and Spanish Pictograph-to-Text translation, respectively. The evaluation metrics used are described in section 9.2.3. The scores obtained are in line with the Dutch 5-gram language modelling-based translation engine, with BLEU reaching a score of 07.32 and 07.18 for English and Spanish, respectively, in the Sclera condition (as compared to a BLEU score of 08.26 for Dutch), and a score of 08.66 and 07.47 for English and Spanish, respectively, in the Beta condition (as compared to a BLEU score of 09.73 for Dutch). The results are satisfactory, considering that Spanish, in particular, has an inflectionally rich morphological paradigm, and many different surface forms could be generated.

10.2.3.3 Beyond the Baseline System

We created a baseline system for English and Spanish, which uses a reverse lemmatiser and beam search decoding on a 5-gram language model. Just like the Dutch baseline system for Pictograph-to-Text translation, it is not able to generate function words, with the exception of articles, and it assumes that the grammatical structure of pictograph languages resembles and simplifies that of a particular natural language. In this section, we describe how our advanced modules for improved Pictograph-to-Dutch translation can be adapted toward other natural languages.

Data-driven machine translation-based methods require a decent amount of training data. Since a parallel corpus of "correctly" written natural language text (English, Spanish, and so on) and pictograph translations does not exist, the first step in creating a data-driven machine translation engine would be to build an artificial parallel corpus, using either one of the two methods proposed for Dutch (see section 9.4.1). This approach requires a suitable monolingual corpus to model the type of content that is created by users with ID, i.e., a corpus includes that includes many references to daily life activities, objects, and emotions, and first and second person forms. One example of this would be the British National Corpus (BNC 2007)[36] for English, or COLA,[37] an orthographically transcribed spoken corpus of youth language, for Spanish. A held-out set can be used for tuning or validation.

The parallel corpus can be used to train a translation model with Moses (Koehn et al. 2007) or to train a neural machine translation engine using the OpenNMT toolkit (Klein et al. 2017). It is recommended to train different versions of the system, using different parameter settings, and possibly also factored information.

10.2.3.4 Conclusion: Pictograph-to-Text Translation for English and Spanish

We have described how the baseline Pictograph-to-Dutch translation system can be extended to other natural languages. To implement new languages, the following components are required: decent connections between the source language's WordNet and Princeton WordNet 3.0, a reverse lemmatiser containing a list of lemmas, inflected forms, part-of-speech tags, and token frequencies, an appropriate training corpus to build the language model, and a new set of parameters to optimise the system's performance.

[36]See footnote 24.
[37]See footnote 25.

10.3 Conclusion: Extending the Translation Technologies to Other Languages

In this chapter, we have demonstrated how a Text-to-Pictograph and Pictograph-to-Text translation tool can be built for pictograph languages other than Sclera and Beta, and natural languages other than Dutch. Our baseline systems are largely language-independent and only require a few modifications, such as language-specific taggers or language models, to enable translation from and into different pictograph languages and natural languages.

We have shown that, in order to build pictograph translation technologies for languages other than Dutch, it is essential to link pictographs to the synsets in that language. Once a pictograph set has been linked to a WordNet in any language, it can easily be transferred to other WordNets through the pre-established links between synsets.

CHAPTER 11

Comparison with Other Pictograph-Based Communication Technologies

In the previous chapters, we described a set of technologies that automatically translate natural language text into pictographs and vice versa, as well as an interface that allows people with limited writing skills to set up pictograph-based messages. The current chapter is dedicated to other applications that also employ pictograph-based technologies to improve communication in situations in which (functional) illiteracy arises, and centers around the differences between these systems and our own translation technologies.

A quick search in the literature reveals that the possibilities are endless. Think of, for instance, efficient pictograph storage and retrieval functionalities, technologies that link pictograph output to speech synthesis software, or tools that help structure the user's environment. These systems do not only consider people with an intellectual disability (ID), but also people who have motor impairments, and people who do not speak the language of their host country.

In the first chapter, we discussed the three main communicative functions of pictographs: crossing language boundaries, enriching the emotional expression, and enabling communication for people with ID. The current chapter provides an overview of pictograph-based technologies for intercultural and augmented communication. We do not consider emoticons, since emoticons are stand-alone image sets that are by default included in most contemporary communication technologies, such as chat clients.

Section 11.1 presents a number of pictograph chat tools that help stimulate intercultural collaboration and personal relations, as an alternative to regular machine translation between natural languages. A more widespread use of pictograph communication technologies lies within the domain of Augmentative and Alternative Communication (AAC). The tools presented in section 11.2 promote the inclusion of people with a physical and/or intellectual disability by giving them a voice through pictograph translation and/or speech synthesis technologies. A large part of this chapter is dedicated to the solutions that have been developed for these users, and their practical or technical limitations, as compared to our own translation technologies.

11.1 Pictograph-Based Technologies for Cross-Cultural Communication

A language barrier occurs between people who don't share the same spoken language (Munemori et al. 2010:473).

One could think that it would be easier to have the whole world learn the same language. Artificial languages, like Esperanto, have met little success, largely due to the initial effort required to learn them. According to Leemans (2001:36), a natural language would probably not work either, because it will never be accepted by some countries, who might worry about cultural domination.

Alternatively, pictograph-based technologies have been proposed as a way to facilitate the communication between people who do not speak the same language. In the following sections, we discuss a number of tools that are developed for non-impaired users. We consider three types of systems: systems that make it simpler to travel abroad (section 11.1.1), educational systems that encourage children to make friends across language barriers (section 11.1.2), and systems that aim at everyday or even professional communication (section 11.1.3).

11.1.1 Travelling

When travelling abroad, communication barriers may arise. A number of pictograph-based technologies address this issue.

The Hotel Booking System is the first known attempt to model pictograph dialogue (Yazdani 1993). Yazdani presents the scenario of a stranger in a foreign city operating a touch screen in the window of a tourist office or a traveller contacting a foreign town's accommodation bureau through a computer. The application lets the user complete his/her booking requirement by selecting pictographs (i.e., number of beds, number of nights, and other special requests), and sends the completed request to the hotel manager. The Hotel Booking System is, to some extent, a precursor of most modern pictograph interfaces.

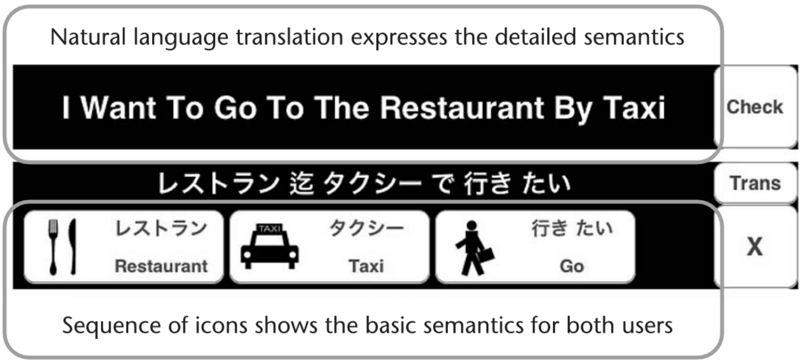


Figure 11.1: The picoTrans interface for Japanese input and English output. Example taken from Finch et al. (2013:33).

The **picoTrans** system (Song et al. 2013) goes one step further and introduces a pictograph-based paradigm for cross-lingual communication on mobile devices. Traditionally, travellers have been able to communicate simple information in foreign countries by the use of large lists of commonly used expressions. These lists or phrase books usually do not allow to express anything that falls outside the scope of the phrase book itself. One possible solution to this problem came in the form of picture-based translation aids, which have existed for some time as paper books. These books contain pages of pictures which represent basic concepts, allowing users to simply point at what they wish to communicate. However, picture books are a let down in terms of their expressive power. This is where modern technology comes in. In the picoTrans smartphone app, the user creates a source sentence by browsing a number of pictograph-based categories and subcategories. Some pictographs may give rise to several lexical surface forms, in which case the user must choose the preferred inflected form. Note that, in our Pictograph-to-Text translation system (see Chapter 9), inflected forms are generated automatically, as our target users have limited reading and/or writing skills. Once the pictograph sequence has been formed, the user is shown the system's suggested source language string for the pictograph sequence. The target language sentence is obtained by translating the source sentence using regular phrase-based machine translation methods for natural languages (see Figure 11.1). In other words, in picoTrans, the pictographs serve as a way to reduce the amount of keystrokes needed to create a source language string, and as a means of standardising the sentences to be translated; the user cannot enter words or phrases that fall outside of the scope of the pictograph interface. This is advantageous for the phrase-based machine translation system, since many translation errors arise from the use of infrequent or misspelled words.

While it is true that the Hotel Booking System and picoTrans have potential for specific communication tasks in foreign countries, such as ordering food, booking a hotel room, or buying tickets, it is important to remark that the pictograph lexicon of both applications is restricted to the domains of transport, accommodation, and a number of other, basic travelling needs. In other words, they do not quite yet pave the way for everyday communication. However, the systems demonstrate that task-oriented pictograph interfaces have great application potential outside the domain of written communication with/by people with ID, as well.

11.1.2 Making Friends Across Language Barriers

The following technologies are designed to prevent stereotyping to different religions, races, and nations by encouraging children to meet people around the world on a personal basis and fostering bonds with each other over dialogue.

PictNet is an online pictograph communication system developed by Takasaki (2006) and Takasaki & Mori (2007), in which each user has his/her own customisable dictionary of approximately 550 pictographs. The user decides the design for every concept in the dictionary (see Figure 11.2) or uses the default pictograph set, and composes messages by dragging the pictographs to a blank canvas. For a more thorough description of PictNet's pictograph interface, we refer back to section 8.1.1. The email client for the PictNet system does not only show the sender's original pictograph message, but also the translated message with the recipient's dictionary, at the same time. This way, the recipients can learn the differences of the pictograph designs through translation and get an understanding of the sender's own cultural background. Note that, as opposed to our Pictograph-to-Text translation system (see Chapter 9), the pictograph strings in PictNet are not translated into natural language text.

Figure 11.2: Pictographs drawn by children to express the concept *good*. Example taken from Takasaki (2006:283).

Munenori et al. (2011) describe a similar tool: **Pictograph Chat Communicator**. The system supports the use of pictograph chat between distant places and contains approximately 500 pictographs that are organised into eight categories. Again, these pictograph strings are not translated into natural language text. In the discussion, the authors argue that the pictographs were not always easy for the users to find, underlining the need for user-oriented interface design. They also remark that the total amount of available pictographs was still insufficient to allow for meaningful communication between participants.

During the development of our static pictograph interface (see section 8.1), we overcame these problems by adopting the principles of user-centred design, and by including a total of 1,660 Beta pictographs and 2,181 Sclera pictographs, based on the most frequently used concepts. Note that even more pictographs can be added in the future, if deemed necessary.

PictNet and the Pictograph Chat Communicator are communication tools that allow children to befriend people from different cultural backgrounds. While their pictograph vocabularies are not yet extensive enough to enable meaningful conversations between participants, both systems prove that basic communication through pictographs is feasible.

11.1.3 Everyday and Professional Communication

A number of pictograph-based systems are aimed at everyday or professional communication.

Leemans (2001) developed **VIL: a Visual Inter Lingua**, a pictograph language (see section 1.3.2.1.2) with its own grammar rules, that is associated with an email client. Following research on pidgins and Basic English, VIL's grammar has a very low complexity. The verb determines which potential arguments are shown on the screen for the user to instantiate: the agent, the source, the theme/object, and the goal/destination. The user first picks a verb, and then fills in its argument slots. Double-clicking a noun pictograph brings up a prompt, in which the user can modify a number of properties about the noun and add adjectives, determiners, or quantifiers.

VIL's pictograph lexicon is restricted to the domain of cooking. Rather than developing a full-fledged application, Leemans' primary objective is to show that it is feasible to meet the need for communication between speakers of different languages by means of pictographs. For instance, he found that it is not harder to encode sentences in VIL than to translate sentences from another unknown language by means of a dictionary.

Note that Leemans assumes that the user masters a certain degree of grammatical awareness in order to identify the verbs, fill in their argument slots, and modify the properties of the arguments. Furthermore, unlike our Pictograph-to-Text translation technology (see Chapter 9), the pictograph messages are not automatically translated into natural language text. With the pictograph lexicon being limited to one domain, namely cooking, it has not yet been proven that a pictograph-based system could be helpful for everyday communication between people who do not speak the same natural language. However, with respect to the design of the pictograph set, the interface, and the grammar, VIL certainly has its merits: We refer back to section 1.3.2.1.2 for the formal characteristics of the pictographs, and to section 8.1.1 for a description of VIL's pictograph hierarchy.

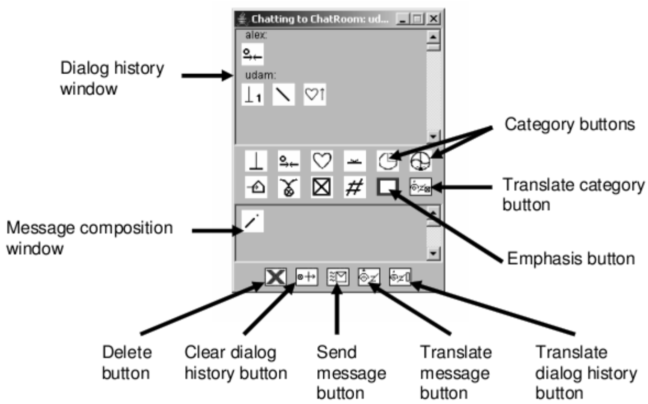


Figure 11.3: The Visual Messenger chat screen. Example taken from Chen (2004:13).

The **Visual Messenger** project by Chen (2004) builds on the idea that visual languages could not only make cross-cultural communication easier, but might also stimulate intracultural communication between speakers of the same natural language.

Just like the VIL system, Visual Messenger was developed with the aim of exploring new communication possibilities. When composing a message, the user selects image categories, which cause associated image window objects to pop up, showing a number of Blissymbols pictographs (Bliss 1965) (see section 1.3.2.1.1) for the user to choose from (see Figure 11.3). The user can choose to display the (textual) labels of the pictographs, but again, the messages are not translated into natural language text.

Fitrianie & Rothkrantz (2005) describe **Lingua** and **ISME** (Icon-based System for Managing Emergencies), two pictograph-based technologies that enable communication in crisis situations, such as acts of terrorism, aviation accidents, or natural disasters. In a crisis event, many people with different cultural origins could be involved, whether they are victims, witnesses, rescue teams, or families, giving rise to the need for a shared language.

In the Lingua system, the user can select a sequence of pictographs as a realisation of an observation of a crisis situation. As opposed to the VIL and Visual Messenger systems, the Lingua translation engine is able to interpret the pictograph sequence, deriving its meaning as a result of the combination of the pictographs, and converts it automatically into natural language text. A parser checks the pictograph sequence against a grammar to determine whether the input is syntactically correct. The grammar rules are modelled after English. The ISME tool, on the other hand, allows a user to report about a crisis situation by placing pictographs on a map where the event occurs. The interface creates a scenario that describes the situation on the map, transforming the coordinate of the pictograph on the map into global coordinates. Road names can be inferred by using a database that contains the global coordinates of all road names.

The Lingua and ISME interfaces assume that the users are familiar with the pictograph grammar at hand, rejecting pictograph sequences that are not syntactically well-formed. In this regard, their architecture resembles our purely rule-based approach toward Pictograph-to-Text translation (section 9.3), which we abandoned due to it having a low recall and its need for resources that are largely developed by hand. One may also argue that, when a real crisis situation ensues, building pictograph messages could be too much of a time-consuming task.

Most of the above-mentioned technologies were devised as a case study to explore new communication techniques, and they do not contain a sufficient amount of pictographs required to meet the objective of facilitating everyday or professional communication between speakers of different natural languages. It has not yet been proven that learning a pictograph language has an actual benefit over learning a new natural language or simply using standard machine translation techniques - at least, when the user in question has no reading or writing difficulties.

11.1.4 Conclusion: Technologies for Cross-Cultural Communication

We have argued that the use of pictograph technology for cross-lingual communication could be beneficial for travellers, especially in situations where specific communication tasks arise. Furthermore, pictograph-based chat tools can be a valuable way for children to meet friends from other countries. Ideally, these applications highlight the subtle differences between different languages and cultures, allowing children to discover other people's traditions or habits. We could expect, however, that people who do not have reading or writing disabilities would rather learn a new language or simply make use of regular machine translation solutions over learning pictograph languages to discuss everyday or professional matters, such as, for instance, business proposals or politics. This research topic could be investigated in the future.

Note that the communication interfaces described above put severe restrictions on a user's freedom of expression by requiring the user to fill in uninstantiated semantic or grammatical slots, and offering only very few pictographs for the user to choose from - at least, for the time being. With many alternative (natural) language technologies available, thorough research is still needed to determine whether pictograph-based communication truly stands a chance of becoming a new standard in online communication.

11.2 Pictograph-Based Technologies for Augmentative and Alternative Communication

In this section, we focus on pictograph-based technologies for AAC. While most of the tools and aids described below are developed for people with ID, we also mention a couple of pictograph-based systems that aim to provide support for people with motor impairments.

This part is structured as follows. Section 11.2.1 presents a number of pictograph-based technologies that do not make use of natural language processing (NLP) techniques to convert pictograph input into written text or speech or vice versa. These systems either display or synthesise the pictograph's (one-word or phrasal) labels, or they do not provide textual output at all. In this regard, they do not fully resemble our own technologies. We then discuss a number of pictograph-based technologies that do make use of NLP techniques, which allows for a better comparison. In section 11.2.2, we present a handful of systems that automatically generate pictographs from natural language text (Text-to-Pictograph translation). The opposite direction is discussed in section 11.2.3 (Pictograph-to-Text translation). Section 11.2.4 summarises the limitations of the currently available technologies.

11.2.1 Pictograph-Based Technologies Without NLP

The tools and devices that are presented in this section do not use NLP techniques to analyse or construct natural language text. However, just like our tools, they aim to facilitate the direct expression of emotions and needs for people who experience language difficulties, either in a face-to-face setting (section 11.2.1.1) or in a remote, online setting (section 11.2.1.2), or they provide help in structuring the user's environment (section 11.2.1.3).

11.2.1.1 Technologies That Give a Voice to People with Speech or Motor Impairments

Many systems allow the user to construct messages by selecting a sequence of pictographs that are associated with textual or phrasal labels. These labels can be read aloud by means of text-to-speech technology.

One of the earliest pictograph-based technologies is the **Minspeak** system (Baker 1982a), which uses the principle of "semantic compaction". A Minspeak board with fewer than 50 keys can produce thousands of spoken sentences. Each key is associated with a pictograph, the meaning of which changes according to the sequence in which it is hit. Messages that are stored in the memory of a microcomputer are retrieved by combining the pictographs (see Figure 11.4). For instance, by associating the *apple* pictograph with the *rising sun* pictograph, the *breakfast* concept can be formed. Note that Minspeak might not be appropriate for people with ID, as most pictographs and their combinations must be learned, and almost never "speak for themselves".

Patel et al. (2004) present the **Image-Oriented Communication Aid**, an interface

Figure 11.4: Minspeak translation of the concept *to sink*. It combines an *opposite* pictograph, a *sheep* pictograph (the sheep jumps, which denotes an *action*), and a *whale* pictograph (a whale *swims*). Example taken from Baker (1982b:4).

intended for preliterate users who require pictograph-based communication devices. Assuming that the end users' utterances are often limited to simple two-word or three-word sequences,[1] the authors propose a two-dimensional, spatially organised image schema. The reasoning behind this design is that a spatially organised image has the ability to express semantic structures and contents of a message that would be lost using linear word ordering, although this argument is not further developed or justified. In the Image-Oriented Communication Aid, the user first selects a verb pictograph, which represents the core meaning of the sentence. The system then displays a semantic template for that verb, which can be filled by selecting the appropriate vocabulary items that fulfill each argument role. The semantic schema also includes sub-roles associated with the main role arguments. For example, the *object* role has the *quality* or *count* sub-roles. The user fills in all the pictograph slots until a sentence is formed. Note that this frame-based method is similar to the one employed in the VIL system by Leemans (2001) (see section 11.1.3). In the discussion, the authors admit that the interface requires some basic level of linguistic and cognitive functioning. However, they also believe that it is less demanding than the functioning required in linear syntactic ordering. These hypotheses are not confirmed, as the Image-Oriented Communication Aid was not tested with real users.

Motocos (Hayes et al. 2010) are AAC devices that support visual communication. Designed for children with an autism spectrum disorder, these portable devices allow the user to communicate by selecting pictographs or by responding to a communicative prompt. Motocos come with a library of pre-installed cards and caregivers are able to add new cards by taking pictures using the built-in camera. The cards can have multiple audio cues assigned to them, which are either recorded with a microphone or synthesised using text-to-speech functionalities. The system is designed for flexibility of communication, either in structured communication settings during

[1]Note that our own target users write an average of 7.7 words and/or pictographs per message (see section 2.2.4.2).

learning activities or for use in unstructured, spontaneous utterances.

There also exist a number of similar, commercial solutions, which are widely used in schools, day centres, and at home. We will not discuss them here, as their core technologies are not disclosed.

11.2.1.2 Technologies That Enable Remote or Online Communication

In this section, we discuss a number of pictograph-based communication systems that are designed for remote or online communication. Their primary objective is to boost an impaired user's e-inclusion by providing tools that help him/her communicate beyond face-to-face settings.

The **Text Messaging with Picture Symbols project** (Müller et al. 2010) evaluates the possibilities for people with communicative disabilities to use text messaging with mobile phones. The devices are accessed by pressing pre-defined pictograph cells on a touch screen. When a message is sent, the pictographs are converted into their textual labels and sent as a regular message to any kind of mobile phone. Incoming text messages can be displayed with pictographs, if the user has the corresponding pictographs stored in the database of the device. The pictographs stored must in that case be named exactly as the words in the incoming text message. In this respect, the system resembles WAI-NOT's original baseline pictograph translation system (see section 2.1).

Messenger Visual (Tuset et al. 2010) is an instant messaging service that allows people to exchange pictographs in real time across the Internet. The program offers a reduced set of 400 pictographs that are classified into 14 categories. Experiments with real users reveal that people with higher communicative skills tend to form more elaborate pictograph messages, whereas people with lower communicative skills make simpler messages that are faster to build and easier to read. Participants were able to communicate with the service, saying that it was both interesting and entertaining.

Keskinen et al. (2012) present **SymbolChat**, a flexible pictograph-based communication platform that produces spoken output using speech synthesis technology (see Figure 11.5). The prototype was evaluated in a field study with the help of nine users with varying degrees of intellectual and motor disabilities. Although the pictograph-based communication was claimed to be fast and fun, participants also thought it was fairly hard. As most users were unfamiliar with the structuring and content of the pictograph set at hand, finding the appropriate pictographs for message construction turned out to be a time-consuming task. Furthermore, after finding the initial

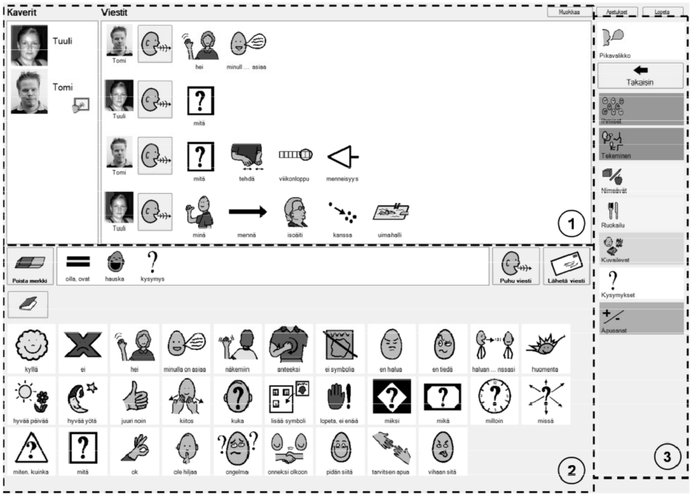


Figure 11.5: The Finnish version of the SymbolChat user interface and an example discussion between two participants. Example taken from Keskinen et al. (2012:281).

pictograph, participants often selected pictographs within the same category without exploring the other categories.

The results of the above-mentioned studies demonstrate that the communication experiences of people with ID can be improved with proper tools that are designed with simplicity and customisability in mind. In Chapter 8, we found that adopting the principles of user-centered design and corpus-based categorisation can lead to satisfactory results with respect to category browsing and pictograph retrieval (see section 8.1.4). Furthermore, our categories can be disabled or enabled by caregivers to accommodate for different needs. It can be expected that prior training and familiarisation with the technologies have beneficial effects over time.

11.2.1.3 Technologies That Structure the User's Environment

A handful of systems provide an aid in structuring the user's environment by replacing or supporting written text with pictographs. We present three noteworthy examples.

Traditional text recipes are often prohibitively difficult to follow for people with language disorders. Tee et al. (2005) developed **VERA**, the Visually Enhanced Recipe Application, which uses a combination of pictographs, animations, audio, and keywords to enhance recipes for people with limited reading skills. Note that this system does not automatically convert textual recipes into the new presentation format. Instead, the application contains a handful of recipes that were manually translated into images by annotators. While evaluations with aphasic users suggest that the combination of visual instructions and navigational structure imposed by VERA may help those with relatively large language deficits to cook more independently, this system was not further developed. One reason for this may be the fact that manually annotating recipes is a time-consuming task, which could be sped up by making use of a Text-to-Pictograph translation technology (see Part II).

Bhamidipaty & Padmanabhan (2007) present **SymAB**, a pictograph-based address book for the semi-literate mobile user. Mobile devices assume a reasonable amount of literacy. The traditional design of the address book as a text-based storage and retrieval interface creates a barrier to its usage for people with limited literacy skills. SymAB proposes a mobile keypad with provision of pictographs, either in addition or as a replacement to the regular alphabet, and a pictograph-based retrieval mechanism of the stored contact entries. Experiments with users with ID revealed that target users could appreciate the enhanced value they perceived from the design, claiming that the address book feature is simpler to use and more comfortable to play around with.

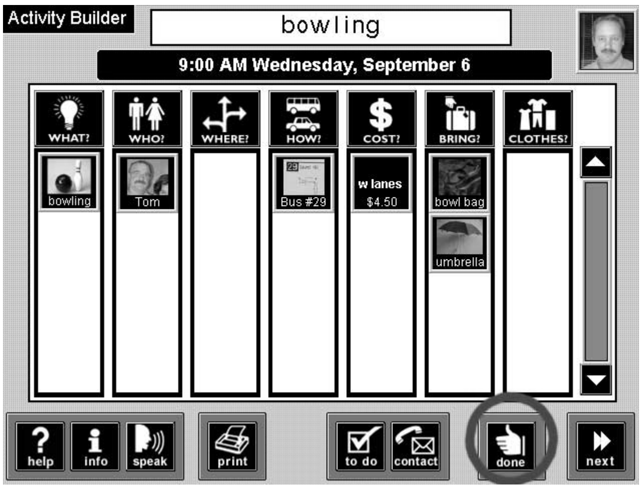


Figure 11.6: A *bowling* activity in Picture Planner. The user needs to make sure that he will bring a bowl bag, an umbrella, and cash money.

Finally, **Picture Planner** (Keating 2006) is a pictograph-driven software interface to provide users with ID with a means to construct and manage activity schedules. The pictographs consists of an image, a textual label, and audio. The planner prompts the user to fill in a number of meta-elements that bear on an activity's successful completion, by asking the user a number of relevant questions, such as "How much money do I need for this activity?" and "How will I get to the activity's location?", among other things (see Figure 11.6[2]). Users of Picture Planner reported an improved self-esteem.

The above-mentioned systems indicate that users with ID may benefit from using computer applications for activity planning or other management tasks, if they are designed with cognitively accessible interfaces.

11.2.2 Technologies That Generate Pictographs from Natural Language Text

Systems that automatically generate pictographs from natural language text, i.e., systems that are similar to the Text-to-Pictograph translation engine (see Part II), are sparse.

Text picturing systems usually implement a processing pipeline, where text is first processed into basic linguistic units, then knowledge is extracted, and finally pictures are found (Poots & Bagheri 2018). Systems that automatically augment written text with pictographs for people with reading difficulties are primarily conceived to improve the *comprehension* of textual content.

Mihalcea & Leong (2008) evaluate the hypothesis that pictorial representations can be used to effectively convey simple sentences across language barriers and to enable communication to and from people with language disorders. For this purpose, the authors introduce the **PicNet** database.[3] PicNet relies on a web-based system for augmenting dictionaries with illustrative images, using volunteer contributions. The images are annotated with a lemma, part-of-speech tag, and synset number according to the WordNet (Miller 1995) lexical database. In this sense, it resembles the design of the baseline Text-to-Pictograph system (see Chapter 3). PicNet can be used to assign pictographs to basic nouns and verbs (see Figure 11.7). However, no pictures are assigned to adjectives, adverbs, or to any other words with a different part-of-speech tag, whereas the aim of our translation system is to cover all content words. Comparative experiments conducted on visual and linguistic representations of information show that a considerable amount of understanding can be achieved by means of pictorial descriptions of nouns and verbs, with results within a comparable range of natural language translations that are obtained by means of statistical ma-

[2] Source of image: http://www.eugeneresearch.org/picture_planner/index.html

[3] Not to be confused with PictNet in section 11.1.2.

chine translation techniques. Note that the system was not tested with people with ID, and that the evaluations, therefore, may not reflect their opinions. Furthermore, the PicNet images - which are a combination of photographs and illustrations - were not specifically designed for the target group, unlike the Sclera and Beta sets.

Figure 11.7: Mixed pictorial and linguistic translation for *You should read this book*. Example taken from Mihalcea & Leong (2008:163).

Widgit Symbols[4] add visual support to printed words. Over the past years, the commercial company Widgit Software has developed a number of tools and applications that make use of the Widgit Symbols set. Their technologies provide teachers and caregivers with a means to construct pictograph-supported text material (**Widgit SymWriter**), or allow users with a reading disability to browse the web independently, by automatically translating words and expressions into pictographs (**Widgit Point**). While the language technology behind Widgit Software's AAC tools is not disclosed, it can be inferred that some form of lemmatisation is applied, as inflected forms are translated into pictographs. With Widgit's pictograph database containing over 14,000 entries, the system often retrieves multiple pictographs for a single word. The lemma *can*, for instance, has one noun (*tin can*) and two verb (*ability* and *permission*) readings, which are all presented to the user in the form of a list. In other words, the system does not make disambiguation decisions like our word sense disambiguation module does (see Chapter 6). Furthermore, unlike our Text-to-Pictograph translation technology (see Part II), the tool does not use semantic relations to display pictographs for words for which no image could be found in the database. For instance, *siamese* (a type of cat) is not translated.

Bautista et al. (2017) present **AraTraductor**, a Text-to-Arasaac translation system for Spanish, using our work on Text-to-Pictograph translation (to which the authors refer extensively) as a source of inspiration. They apply lemmatisation and part-of-speech tagging, and their system is also able to detect multi-word expressions for the generation of complex pictographs (see section 1.3.2.3.1). If the user clicks a pictograph, and there is more than one possibility for it, a list of variants is shown. In other

[4]http://www.widgit.com/

words, the system does not yet calculate an optimal translation, and it does not yet use the semantic relations between concepts, such as synonymy, to find pictographs for words that have no pictograph associated to them. AraTraductor is currently a work in progress.

The above-mentioned systems provide a direct link between lemmas and pictographs, but syntactic analysis and semantic processing is kept to an absolute minimum. While PicNet uses WordNet, it does not use WordNet's semantic relations between concepts.

Poots & Bagheri (2018) consider a number of technologies that automatically annotate text with pictures or 3D images, but they are not specifically designed for AAC purposes. Joshi et al. (2006) describe an unsupervised approach for automatically adding pictures to a story. They extract semantic keywords from a story and search an annotated image database, but do not try to translate the entire story. WordsEye (Coyne & Sproat 2001) is a 3D scene rendering system, which parses text into a semantic representation, and, in turn, creates a 3D scene. It augments existing 3D software tools by, for instance, allowing a designer to quickly set up a scene to be later refined by other methods. Finally, Agrawal et al. (2011) propose an algorithm to match sections of a school textbook to Internet images.

11.2.3 Technologies That Generate Natural Language Text from Pictographs

Pictograph-enabled writing tools allow users to write (and synthesise) natural language text using only pictographs, or a combination of text and pictographs. The tools that are presented in this section make use of NLP methods to build grammatically correct sentences from compressed and incomplete input. Systems that automatically translate pictographs into natural language text for people with writing difficulties, i.e., systems that are similar to the Pictograph-to-Text translation engine (see Part III), are primarily conceived to facilitate the *construction* of textual content, either for communicative purposes (section 11.2.3.1) or for querying the Internet (section 11.2.3.2).

11.2.3.1 Writing with Pictographs

There are many examples of technologies that automatically convert pictograph input into natural language text by means of rule-based linguistic modules that expect "grammatically correct" pictograph input. Note that most of the below-mentioned systems put severe restrictions on the users' freedom of expression, either by requiring them to adhere to a fixed pictograph order according to some pictograph grammar, or by prompting them to fill in a number of uninstantiated role slots. This contrasts with our approach(es) toward Pictograph-to-Text translation (see Chapter 9), since we allow the users to freely insert (incomplete or ungrammatical) pictograph sequences. While the output may not always be fluent, all pictograph input in our system is used during the analysis.

Blisstalk (Hunnicutt 1986) is an electronic communication board on which Blissymbols are displayed. Each of the 1400 standard Bliss pictographs corresponds to an entry in the lexicon, containing the pronunciation of the item, its part-of-speech tag, and other features, such as number and tense. There exist (at least) three different grammars which are associated with the use of Blissymbols. The "telegraphic" grammar omits function words and pays less attention to the word order. In this case, no syntactic analysis is used for the conversion into natural language text. The second grammar uses linguistic knowledge and expects natural spoken language ordering as input. The third grammar, also known as the "Bliss syntax", was developed by Bliss (1965). This syntax is also used in the **Blissvox** system for Hungarian (Olaszi et al. 2002) - even though the grammar rules of the Bliss syntax cover syntactic structures that would be strange for Hungarian. For instance, Hungarian is classified as an SOV-type language, whereas Bliss syntax only accepts SVO-type input. The task of the linguistic module is to transform the sequences of words into grammatically well-formed Hungarian sentences with inflected word forms.

The **Prothèse Vocale Intelligente** (PVI) system (Vaillant 1998) for French stems from hands-on research involving people with cerebral palsy. This disorder involves permanent lesions of the central nervous system, causing neuromotor problems and cognitive deficits, which are manifested in difficulty with syntax and a preference for short, telegraphic-style communication. A PVI user can select pictographs from the grid by clicking, by using a switch (for limited motor abilities), or by tapping a keyboard (see Figure 11.8). Once composed, pictograph strings are passed to an analysis module, which translates the sequences into a meaning representation. The central idea behind this module is to build up the semantic relations between predicates and attributes. The first phase consists of spotting the predicative pictographs, i.e., those which have selectional features, such as verbs. The system then moves on to the remaining pictographs and picks out the best candidates for each of the predicate's argument slots. Secondly, a lexical choice component, which maps icons onto French words, prepares a conceptual graph, and adds uninstantiated morphosyntactic variables where appropriate. Finally, the linguistic conceptual graph is transformed into one or multiple French sentences using a tree-adjoining grammar-based generator (Joshi et al. 1975). Vaillant (1998) recognises that the fragile balance between expressive power and interpretation accuracy is a remaining central problem in the PVI approach, as the selectional features impose severe restrictions. For instance, while ellipses in pictograph input occur frequently, the system is unable to interpret incomplete sequences. Note that we chose not to continue the development of a purely rule-based Pictograph-to-Text translation engine due to similar observations (see section 9.3). PVI has not been extended to other languages. This may be due to the fact that the system relies on a visual language mapped specifically to French words, and requires a language-specific grammar for analysis, as well as a language-specific tree-adjoining grammar for generation.

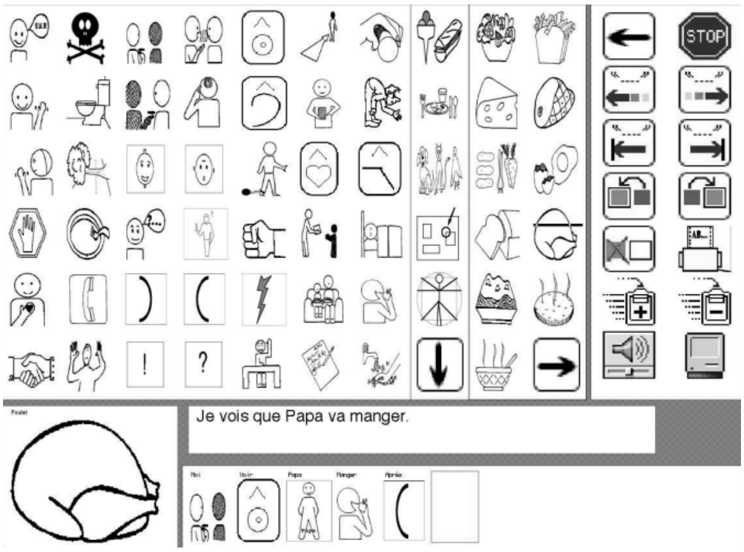


Figure 11.8: The PVI interface. Example taken from Vaillant (1997:188).

Bhattacharya & Basu (2009) developed **Sanyog**, a pictograph-based communication aid that converts pictograph input into grammatically correct sentences in Bengali or Hindi. The original design of the Sanyog system made use of the *compansion* approach (McCoy et al. 1998). Compansion is a three-stage process. The first stage is word order parsing, which groups input words into sentence-sized chunks. In the second stage, the semantic parsing phase, a semantic representation is created by instantiating a semantic frame with the input sequence. The final stage is the generation of a sentence from an instantiated semantic frame. This includes inserting prepositions and determiners and adding morphological modifications. However, for most Indian languages, including Bengali and Hindi, an application of this approach was not possible at the time, due to the absence of the necessary linguistic knowledge bases, such as a FrameNet and WordNet. For that reason, Bhattacharya & Basu developed the QR (query-response) model. Rather than giving the user full control over the composition of pictograph sequences, the QR model expects users to select a verb and answer a series of queries relating to the roles associated with the semantic frame of the verb. Through semantic slot filling, the system ensures that the input will be converted into a grammatically correct sentence. Sanyog is currently deployed at a number of institutions in India.

Yet another system that ensures the syntactical and semantic correctness of the input is the **Simple Upper Ontology (SUpO)** (Martínez-Santiago et al. 2015), an ontology with linguistic knowledge used to model the usual language of beginning communicators, which lies at the base of their Pictogrammar technology. SUpO is a semantic grammar that takes FrameNet as a starting point. It covers a controlled language made up of 621 words and 189 different kinds of grammatical roles and sentences in English and Spanish. The authors note that the development of the SUpO ontology is an iterative exercise, in which frames have to be carefully adapted in order to avoid under- and over-specialisation, which may lead to the generation of incorrect sentences at the syntactic and/or pragmatic level.

SymbolPath (Wiegand & Patel 2012) targets individuals with severe speech impairments who have concomitant limb impairments. SymbolPath presents a grid of pictographs which are grouped based on their lexical roles (actors, verbs, objects, and modifiers). To create a message, the user creates a continuous path through a set of pictographs. The only requirement is that a continuous path is drawn through all desired items without breaking contact with the interface. Once the user breaks the path, the language module attempts to concatenate a meaningful and syntactically accurate utterance. The language module needs to deal with two issues. First of all, the path may contain both target elements and irrelevant elements, meaning that the superset must be pruned to select the most likely candidate pictographs. Secondly, the system might have to re-order the pictographs. To deal with these issues, SymbolPath uses semantic frames, semantic *n*-grams, and the characteristics of the path. The system attempts to find predicative pictographs (i.e., those which have selectional features) in the selected path that can be filled by a set of relational items. Semantic *n*-grams are used to assign each potential utterance a value that corresponds to the probability of that combination of words appearing in a sentence together, regardless of order. Finally, the path characteristics determine that pictographs that collide with a larger area of the user's drawn path should be assigned a greater likelihood than icons that are only marginally on the drawn path.

Figure 11.9: User Interface of Symbol Dragoman in Arabic. Example taken from Ding et al. (2015:2).

Lastly, **Symbol Dragoman** (Ding et al. 2015) is a web-based application that aims to allow AAC users to communicate in Arabic or English. Taking short pictograph sequences as input, the system constructs one or more meaningful and grammatically well-formed natural language sentences. For the generation of English, the system uses the keywords that are attached to the requested pictographs to search through a large, indexed sentence corpus; the order of the pictographs does not matter. In a way, it is similar to the PRESEMT approach (see section 9.4). It is interesting to note that Symbol Dragoman allows Arabic users to work in both languages, with the pictographs and words appearing from right to left for Arabic and from left to right for English. As compared to the previously discussed approaches, this tool offers its users more expressive freedom in terms of pictograph message composition, although the authors note that more satisfactory results could be obtained by expanding the

Arabic sentence corpus.

Related to Pictograph-to-Text translation is the domain of image captioning, i.e., the process of generating textual description using techniques from NLP and computer vision. The main objective of image captioning is to learn presentations of the interdependence between concepts in the image and the subsequent creation of a concise, sentential narration (Mullachery & Motwani 2018). The image captioning task consists of an encoding and a decoding step (Yang et al. 2017). During the encoding step, the image is encoded into a target vector. This can be achieved by means of state-of-the art techniques such as neural network models (Vinyals et al. 2015), which represent the image as a single feature vector using the top layer of a pre-trained convolutional network, or attention-based models (Xu et al. 2015), which use the vector made up by the representations of the image's subregions as the source vector. In the decoding step, sentences are generated by decoding the target vector. This is usually done by means of long short-term memory-based recurrent neural networks. An in-depth study of this topic is beyond the scope of this work.

11.2.3.2 Web Searching with Pictographs

Maiti et al. (2011) discuss the advantages of pictograph-based representations of web information. Paintings, illustrations, and photographs can be used as a medium to represent web information, helping illiterate people to get some basic information from a web page. To compose a query and to fire it in search engine, the authors propose a pictograph-based interface. Just like in our translation systems, each pictograph in the query is mapped onto a number of synonymous words. Once the query is fired, the search engine retrieves the first ten web pages, downloads the source code of these pages, and filters out extra information and noise. The system attempts to find matching patterns in the paragraphs of the documents, and fills in an answer template using pictographs. For instance, the query *culture + Assam*[5] generates a number of pictographs that are related to Assam's religion, its products, and its writing system. In other words, the system does not only translate pictographs into text for querying web documents, but also translates text back into pictographs for visualising the retrieved information to the user. While this system is just a simple case study in a very limited domain (i.e., tourism), pictograph-based web browsing is a compelling topic that should be investigated in future work.

[5] Assam is a mixture of Mongolian, Indo-Burmese, Indo-Iranian, and Aryan culture.

11.2.4 Conclusion: Technologies for Augmentative and Alternative Communication

We presented a number of pictograph-based applications and devices that are specifically developed for people with a disability. We discussed a number of speech-generating devices that facilitate local, face-to-face communication for people with motor impairments. Pictographs in these devices replace entire words or concepts with a primary aim of reducing the number of keystrokes that would be needed in order to construct the sentence to be synthesised, but they do not use NLP techniques to translate the pictograph sequences into natural language text. Furthermore, most of these devices require the user to select an inflected word form after selecting a base concept. As such, they are less suited for people with reading disabilities, who might experience difficulties in picking an appropriate form. Other systems are specifically developed for preliterate users, but they require the user to select a predicate and fill in its semantic template. We have argued that this approach requires a certain degree of grammatical awareness. Furthermore, these systems limit the user's freedom of expression by expecting grammatically or semantically complete input. The pictograph-based systems for remote communication that do not employ NLP techniques to translate text into pictographs or vice versa prove that, indeed, people with cognitive and communicative disabilities can experience increased participation by means of accessible communication tools. We have also argued that technologies that are primarily designed to facilitate planning and item retrieval, such as the VERA system for recipes or the Picture Planner, may improve a cognitively impaired user's self-esteem and independence.

Systems for remote communication that use NLP techniques to translate natural language text into pictographs are sparse. While shallow linguistic analysis methods are used, the systems described above do not use the semantic relations between concepts and they either do not translate the whole sentence, or they do not employ syntactic or semantic taggers for deep linguistic analysis.

There exist a number of systems that translate pictographs into natural language text using NLP techniques. However, most of these language generation tools expect grammatically or semantically complete pictograph input, and they are not able to generate natural language text if not all the required grammatical or semantic roles are provided by the user. This input method could be inappropriate for users who do not have the linguistic skills necessary to break down their intended message into semantic or syntactic units. For instance, when a user is prompted to create a message by selecting a predicate or a verb, he/she first need to realise or understand what a verb is, and who or what the arguments of that verb are. Most of the above-mentioned studies do not include real end users in the development or evaluation of the translation tools. Therefore, further research is needed to determine whether these kinds of input methods are truly helpful for the target group, or whether they were simply devised to avoid the generation of ungrammatical text.

Finally, it is also interesting to note that all of the above-mentioned solutions focus on just one natural language, with the sole exception of the Widgit Software products and Symbol Dragoman. Most importantly, extending the systems to other languages would require the developers to manually establish new connections between the pictographs and words in those languages, which is a time-consuming process. This contrasts with the largely language-independent design of our translation tools and the use of interlingual WordNet links, which facilitate the extension of our technologies toward other languages (see Chapter 10).

Conclusion and Future Work

In this dissertation, we have presented tools that automatically translate natural language text into pictographs and vice versa for people with an intellectual disability (ID), allowing them to read and write status updates, emails, and chat messages in online environments. While we focussed on the language pair Dutch to Beta/Sclera and vice versa, we developed methods that are as language-independent as possible. During the development and improvement of the translation engines, we made use of various existing tools for natural language processing, such as a syntactic parser and machine translation toolkits, and we developed the following tools and methods:

1. A stand-alone spelling corrector that is tailored toward the characteristics of text written by people with ID;
2. A syntactic simplification (and compression) module that automatically simplifies natural language text as a pre-processing step for pictograph translation;
3. A temporal analysis module that analyses the temporal characteristics of the input sentence and generates an appropriate temporality indicator;
4. A method to incorporate word sense disambiguation into the Text-to-Pictograph translation pipeline;
5. A static pictograph hierarchy, designed according the principles of user-centered design, that allows people with ID to construct pictograph-based messages;
6. A pictograph prediction tool that suggests relevant pictographs to the user, based on the previously selected pictographs;
7. Language modelling-based and data-driven machine translation-based methods toward Pictograph-to-Text translation.

Due to the fragmented nature of our research, general conclusions are not easy to make. Therefore, we first discuss our contributions to the different subtasks in more detail, and we summarise our experiments and the results. We then present our personal observations and recommendations with respect to the development of language technologies for people with ID, as well as several possibilities for future work.

Text-to-Pictograph Translation

For the development of the Text-to-Pictograph translation technology, we took the baseline translation system by Vandeghinste et al. (2017) as our starting point (see Chapter 3). We evaluated this system using automated metrics, manual assessments, and focus groups with real users, and found that the technology could be improved in (at least) three different ways.

The first improvement concerns the development of a context-sensitive spelling correction tool that is tailored toward the characteristics of text written by people with ID (see Chapter 4). Using a parallel corpus of 1,000 messages written by people with ID and their manually corrected forms, we were able to extract a large set of character sequence rewrite rules, which reflect the phonetic confusion (i.e., the orthographic approximation of a word's pronunciation) that often causes target users to misspell a word. In the spelling correction tool, these rewrite rules are applied to generate variants for non-existing words or infrequent real words. Once all spelling variants are retrieved, a character-based fuzzy matching technique is used to find the best combination of variants and to perform additional character substitutions when a strong context match is found. We have shown that our approach toward context-sensitive spelling correction outperforms more conventional methods, such as beam search decoding on language models. However, there are still some points to improve. The system struggles with the normalisation of words that contain typographic as well as phonetic errors, and the variant generation module has not yet been optimised. Furthermore, we should measure the effects of using more parallel data, allowing us to extract more character sequence rewrite rules, and the use of different corpora for fuzzy matching. Nevertheless, we have succeeded in developing a methodology toward spelling correction that only requires a small set of manually corrected text, making our approach appropriate for other languages or normalisation tasks for which very few or no parallel data are available.

The second improvement is a syntactic simplification (and compression) module, which automatically simplifies natural language text as a pre-processing step for pictograph translation (see Chapter 5). The aim of this module is to obtain more consistent, shorter, and more comprehensible translations, as opposed to the almost *verbatim* pictograph-per-content word translations that were generated by the baseline system. With no parallel corpus of unsimplified and simplified text at our disposal, we opted for a rule-based approach that makes use of syntactic parsing. By using recursion and applying the simplification operations in a manually pre-defined way, only one syntactic parse is needed per message - in this sense, our syntactic simplification module distinguishes itself from other rule-based systems that were previously developed for less resourced languages. Using automated metrics and manual assessments on newspaper text and user-generated content, we are able to report excellent results, whereas our user tests with teachers, caregivers, and parents reveal that simplified pictograph sequences are either perceived as "less difficult", or interpreted more accurately than non-simplified sequences. The simplification tool is augmented with an optional module for verb group simplification and temporality detection, which allows more experienced pictograph users to recognise the temporal properties of a pictograph sequence. While the syntactic simplification system is not developed for generating textual output, but for generating pictograph output, with simplified text as an intermediary phase, it paves the way for the creation of a text simplification tool for Dutch, the first of its kind. Note that, in order to develop such a system, lexical simplification should also be taken into account (Bulté et al. In press).

The third improvement is a word sense disambiguation module, which (partly) makes up for the lack of proper semantic processing in the baseline system (see Chapter 6). We implemented an existing tool into the translation pipeline, adding the word sense disambiguation scores as features of the synsets which are attached to the input tokens. We adapted the A* path-finding algorithm, which calculates an optimal pictograph sequence for a given input message, to include the WSD score, biasing the selection of the pictograph toward the winning sense. As compared to the baseline system, which used the most frequent sense of each word, the system with word sense disambiguation is now less likely to pick the wrong pictograph for an ambiguous word.

We evaluated the complete, improved Text-to-Pictograph translation pipeline with experienced pictograph readers and found that participants consistently provided more accurate and more adequate transcriptions for the pipeline pictograph translations than for the baseline pictograph translations (see Chapter 7).

Note that, albeit a time-consuming task, significant improvements can be made by adding even more pictographs to the lexical-semantic database. Since the Text-to-Pictograph translation tool does not cover the complete set of Sclera and Beta pictographs, the lexical coverage of the tool can still be improved. This observation relates to the idea of using a different lexical-semantic resource, such as ConceptNet (Speer & Havasi 2012), which uses automatically generated representations of word meanings as vectors (see Chapter 2). The potential added value of using word embeddings over WordNet synsets within the context of pictograph translation is a topic worth investigating in future research.

Pictograph-to-Text Translation

The Pictograph-to-Text translation engine relies on (a combination of written text and) pictograph input (see Chapter 8). Following the principles of user-centered design, and building on the work by Daems et al. (2015), we developed a static pictograph hierarchy based on a large corpus of user-generated content, which we used to infer linguistic knowledge and to create an interface that is largely adapted toward the vocabulary and interests of our end users. Using an existing tool for topic detection, the semantic relations between concepts, and frequency information, we proposed a three-level hierarchy that covers a total of 1,660 Beta pictographs and 2,181 Sclera pictographs. During hands-on sessions with the end users, limitations and expectations were discussed, and the hierarchy was put to the test. The new interface was found to be a major improvement over the old category system that was used on the WAI-NOT website. In addition to the static interface, we developed a dynamic pictograph prediction tool, which suggests contextually or semantically relevant pictographs to the user, based on the previously selected pictographs. We built one model that relies on the pictograph's immediate context or n-gram information, and another model that relies on word associations in a context that is broader than n. Our evaluations show that the n-gram model outperforms the association model, and that combining both models does not lead to significant improvements. The performance of the prediction tool can be considered satisfactory, given that the context of the messages is often too narrow to make informed decisions. The tool could be improved by training it on individual users and inferring their personal preferences over time. However, caution must be exercised when venturing into the domain of user-based logging, due to privacy issues.

For the translation of pictograph input into natural language text, we considered various approaches toward the challenge of generating rich output from underspecified input (see Chapter 9). The baseline system for Pictograph-to-Text translation generates natural language from pictographs using either n-gram language models or long short-term memory language models. In the n-gram-based approach, the system performs beam search decoding on a 5-gram language model. In the long short-term memory-based (LSTM-based) approach, the system re-scores a hypothesis list using LSTM language models. We were able to measure significant improvements using the n-gram based approach, and found that using small corpora that appropriately model the language of our target users leads to better performance than using large mixed-domain corpora. For the data-driven machine translation approaches toward Pictograph-to-Text translation, we constructed a parallel corpus using the Text-to-Pictograph translation engine on a monolingual corpus of Dutch written text. In the phrase-based machine translation method, a translation model is learned from the parallel corpus, whereas in the neural machine translation method, a neural machine translation engine is trained. Using automated metrics, we were able to show the added value of data-driven machine translation approaches over language modelling-based approaches. More specifically, significant improvements in BLEU were measured when using the neural machine translation system. Based on adequacy and fluency ratings, and pairwise ranking by human annotators, we were able to confirm that neural machine translation is currently the most effective approach toward Pictograph-to-Text translation, even though we have not yet exploited the potential of neural machine translation - a field that is constantly evolving - to the best possible extent. We did not consider rule-based approaches, as their recall is often too low, and building a rule-based system is a very time-consuming task.

As opposed to other Pictograph-to-Text translation systems, which expect users to adhere to a fixed pictograph syntax or require users to fill in semantic role slots, we allow our users to freely express themselves by putting no constraints on the order or amount of pictographs allowed. Although, this way, the output may not always be fluent or grammatical, our translation engine always attempts to find the best possible translation for any given pictograph string.

General Observations and Recommendations

Since the request for specialised language technologies came from the environment of the end users themselves, involving the target group and their environment during the development of the translation engines has proven to be essential for the success of this research project. At various points during the creation of the tools, we were confronted with a dire need for user-generated content, in vivo evaluations, and personal opinions, and we are convinced that certain technological decisions could not have been made without consulting our stakeholders. For the development of applications that are truly tailored toward the needs of the end users - making them less likely to be abandoned in an early stage -, creating meaningful collaborations with the user's direct environment (i.e., non-profit disability organisations, special needs schools, day centres, and specialised research groups) is crucial. While this recommendation may seem obvious, there are very few examples in the literature of pictograph-based technologies that involve end users for the evaluation of the tools, let alone technologies that involve them from the very beginning of the project. As a result, the large majority of these technologies end up on the shelf.

In practice, involving target users can be difficult for several reasons. For instance, testing is a particularly long process, especially since it requires very small steps to be taken. However, if that time is spent meaningfully, and a solid methodology is adopted, the feedback obtained can be used to make appropriate modifications, which lead to better results on the long term. A second difficulty relates to privacy issues. When collecting user-generated content, logging information, or involving people with ID in user evaluations, their informed consent must be obtained, and for ethical reasons, (meta)data should be anonymised. Collaborating with people with ID requires an inclusive approach that affirms the importance of respecting their voices and perspectives. Related to this observation is the fact that one size does not fit all. Different needs and abilities require different, tailor-made solutions. Taking into account a wide range of skills and disabilities during the development of the pictograph translation technologies, we created a number of optional modules and settings that can be fully customised to a user's specific needs. Examples of modules that can be (de)activated, if deemed necessary, are the spelling corrector, the syntactic simplification module, the compression module, the temporal analysis module, and the different (sub)categories of the static pictograph interface.

With the pictograph translation technologies, we have contributed to the inclusion of people with ID in online environments. While the movement toward full inclusion of people who do not have the capability to use modern information technology is just beginning, an increasing number of software companies and research groups are attempting to close the gap between the technology-empowered communities and the technology-excluded communities. This challenge is also reflected in the European Union's digital inclusion policy, which particularly addresses people with physical and cognitive disabilities, the economically inactive, immigrants, and the elderly.

Ongoing and Future Work

With the development of the Text-to-Pictograph and Pictograph-to-Text translation technologies, we have delivered various sub-modules which show potential for application in other domains: a methodology toward context-sensitive spelling correction for normalisation tasks for which very few or no corrected data are available, a syntactic simplification module that could be adapted for generating textual output, thus giving rise to the first syntactic simplification engine for Dutch, and so on. Although various approaches were considered for each of the sub-modules, optimisation of some of the techniques used is still desirable. We presented these limitations in the conclusions of the relevant chapters.

We have paved the way for applications and technologies that promote the (digital) inclusion of excluded communities; future work on the pictograph translation technologies is most likely to be directed toward new groups of target users.

For instance, for immigrants who are in the process of learning the language of their host country, pictographs can be used as a stepping stone toward a better understanding or expression of foreign languages. One possible solution to this problem came in the form of pictograph books, allowing non-native speakers to point at images and communicate their basic needs. In March 2018, we received funding to create basic pictograph translation technologies for immigrants living in the Brussels Periphery. For this purpose, the pictographs from the Pictogrammendatabank,[6] a collection of images covering various aspects of daily life and concepts that are related to administration and integration, were linked to synsets in the Cornetto database. We are currently digitising the pictograph books of the Pictogrammendatabank in the form of a Pictograph-to-Text translation technology, which converts pictograph sequences into natural language sentences, helping immigrants express themselves without feeling exposed when looking up a word or a phrase. Conversely, a better comprehension of the foreign language at hand can be achieved by means of the Text-to-Pictograph translation technology. The tool can be used by employees in the Brussels Periphery to supply additional, visual support to foreign clients or customers.

Another potential target group is (elderly) people with aphasia. People with aphasia often experience difficulties in understanding the speech of other people. Their need for specialised materials varies from person to person. Since images reduce the demand on a user's cognitive skills, pictograph translation technologies can play a

[6]https://www.derand.be/pictogrammendatabank

crucial role in facilitating their production and understanding of natural language. An example scenario would be as follows. A user might struggle with finding the words he/she want to say. An assistive technology, such as a smartphone app, detects the hesitation and offers a number of suggestions. The user can complete his/her sentence using pictographs, which are either suggested by the pictograph prediction technology or manually selected in the pictograph interface, or a combination of pictographs and speech. The resulting sequence is transformed into natural language text by means of the Pictograph-to-Text translation engine and read aloud by a text-to-speech module.

Other users who may benefit from pictograph-based technologies are people with limited motor skills, who can use the pictograph interface in combination with specialised input devices to communicate their ideas more effortlessly, and people with low literacy skills.

In conclusion, our pictograph-based AAC technologies can offer people with reading, speaking, or writing difficulties a temporary or permanent form of support, helping them live a more independent life.

With this dissertation, we hope we have inspired you to contribute to the eradication of e-exclusion.

Appendix A

Abbreviations

This list contains the abbreviations.

AAC	Augmentative and Alternative Communication
AAIDD	American Association on Intellectual and Developmental Disabilities
BRNN	Bidirectional Recurrent Neural Network
CGN	Corpus Gesproken Nederlands (Corpus of Spoken Dutch)
ID	Intellectual Disability
ISME	Icon-based System for Managing Emergencies
ISO	International Organisation of Standards
LSTM	Long Short-Term Memory
MCR	Multilingual Central Repository
MT	Machine Translation
NLP	Natural Language Processing
OOV	Out-Of-Vocabulary
PSET	Practical Simplification of English Text
PVI	Prothèse Vocale Intelligente
SAL	System for Augmenting Language
SOV	Subject-Object-Verb
SVO	Subject-Verb-Object
VCM	Visualisation des Connaissances Médicales
VIL	Visual Inter Lingua
WSD	Word Sense Disambiguation

APPENDIX **B**

Development Set and Test Set

B.1 Development Set

Original development set	**Corrected development set**
Hoe is het met je broers?	Hoe is het met je broers?
Is *name* jouw liefje nog?	Is *Name* jouw liefje nog?
name je schrijf niets waarom en wat heben mji zusen gechreven ik zie u op maandag op shool oke	*Name* je schrijft niets waarom en wat hebben mijn zussen geschreven ik zie u op maandag op school oké
Dag zus alles goed met jou?	Dag zus alles goed met jou?
ik vond het gisteren heel leuk groetjes	Ik vond het gisteren heel leuk groetjes
name nu zaterdag is het zwemmen groedjes van *name*	*Name* nu zaterdag is het zwemmen groetjes van *Name*
Name, ik drink aalleen maar koffie en water, voor de rest geen frisdrank, geen wijn, geen bier, niets!	*Name*, ik drink alleen maar koffie en water, voor de rest geen frisdrank, geen wijn, geen bier, niets!
Ik word dus niet snel zat, en jij?	Ik word dus niet snel zat, en jij?
drink je soms wat, ik bedoel alcohol, maar jij rookt!	Drink je soms wat, ik bedoel alcohol, maar jij rookt!
ik moet nog veel rusten ik moet verschlek nog vier of fijv weeken int siekenhuis blijven	Ik moet nog veel rusten ik moet waarschijnlijk nog vier of vijf weken in het ziekenhuis blijven
Ik ga het nog niet vertellen, je zal het morgen zien tijdens sport en spel!	Ik ga het nog niet vertellen, je zal het morgen zien tijdens sport en spel!
ja ik weet het *name* je moet niet verdrietig zijn ze gaat wel weer vriendin met je zijn	Ja ik weet het *Name* je moet niet verdrietig zijn ze gaat wel weer vriendin met je zijn
kom je morgen naar school?	Kom je morgen naar school?
en juf *name* was *name* vandaag op school en *name* heb probleem met haar vader	En juf *Name* was *Name* vandaag op school en *Name* heeft problemen met haar vader

Hallo *name*, heb je veel verdriet?	Hallo *Name*, heb je veel verdriet?
Was het een oude hond?	Was het een oude hond?
oke meester *name* maar wil jij mee duimen voor mijn broer want hij was wel een beetje zeenuwachtig oke groetjes	Oké meester *Name* maar wil jij mee duimen voor mijn broer want hij was wel een beetje zenuwachtig oké groetjes
oke moet ik ook aardbei mee brengen?	Oké moet ik ook aardbeien meebrengen?
meester ik beloof dat ik niet ga roepen	Meester ik beloof dat ik niet ga roepen
praat maar met *name* die is mijn vriendin niet en ook *name* die is ook mijn vriendin niet	Praat maar met *Name* die is mijn vriendin niet en ook *Name* die is ook mijn vriendin niet
En ik heb al vier tekeningen van kangoeroes aangekregen.	En ik heb al vier tekeningen van kangoeroes aangekregen.
Dus mijn muur begint heel vol te hangen.	Dus mijn muur begint heel vol te hangen.
straks zijn het de rode duivels	Straks zijn het de Rode Duivels
hij zit weer de kussen aan mijn hand	Hij zit weer te kussen aan mijn hand
bedankt voor de mooie kaart *name*!	Bedankt voor de mooie kaart *Name*!
hallo *name* alles cava he ga jij morgen snoep meenemen op schoolreis ik wel	Hallo *Name* alles goed hé ga jij morgen snoep meenemen op schoolreis ik wel
Ik zal eens kijken of ik je kan toevoegen op Facebook.	Ik zal eens kijken of ik je kan toevoegen op Facebook.
vandaag is het koud buiten.	Vandaag is het koud buiten.
doet je teen nog pijn?	Doet je teen nog pijn?
dag *name*, ik vind dat heel lief van jou dat je een e-card hebt gestuurd naar mij	Dag *Name*, ik vind dat heel lief van jou dat je een e-card hebt gestuurd naar mij
Name, ik wens je een goed weekend!	*Name*, ik wens je een goed weekend!
ben jij aan het slapen	Ben jij aan het slapen
dag *name* ik heb een brobleem ik heb tegen *name* gezegd dat ik geen vrienden wil zijn	Dag *Name* ik heb een probleem ik heb tegen *Name* gezegd dat ik geen vrienden wil zijn
maar hij blijft berichtjes sturen wat moet ik doen altijd nee zegen met de duim	Maar hij blijft berichtjes sturen wat moet ik doen altijd nee zeggen met de duim
Dag *Name* moet ik een bord spaghetti op de grond gooien mijn mama moet ook een bord spaghetti op de grond gooien	Dag *Name* moet ik een bord spaghetti op de grond gooien mijn mama moet ook een bord spaghetti op de grond gooien
ik ben blij dat ik dinsdag bij jou mag slapen op school.	Ik ben blij dat ik dinsdag bij jou mag slapen op school.
sport morgen en dan moet jij een beetje duner worden oke ik ga morgen ook sporten	Sport morgen en dan moet jij een beetje dunner worden oké ik ga morgen ook sporten
ik ben vandaag naar de dieren geweest met mijn mama en mijn grote zus en er waren veel dieren	Ik ben vandaag naar de dieren geweest met mijn mama en mijn grote zus en er waren veel dieren
hallo ik ben het *name* en ik wil zeggen dat ik luizen heb	Hallo ik ben het *Name* en ik wil zeggen dat ik luizen heb
mijn mama heeft gezegt vanmorgent gezegt tegen mij dat je rond half 2 mag komen	Mijn mama heeft gezegd vanmorgen gezegd tegen mij dat je rond half 2 mag komen

want ik moest u iets geven woensdag maar dat zul je wel de woensdag wel zien	Want ik moest u iets geven woensdag maar dat zul je wel de woensdag wel zien
ken je en condoo ja of nee	Ken je een condoom ja of nee
name kijk op mijn wai-not en klik op mijn foto dan ga jij zien welke hond krijg ik	*Name* kijk op mijn Wai-Not en klik op mijn foto dan ga jij zien welke hond krijg ik
Hoe is het met je niewe bril, zie je nu beter?	Hoe is het met je nieuwe bril, zie je nu beter?
Bedank je ouders voor hun lieve brief!	Bedank je ouders voor hun lieve brief!
oke ik ben jouw vriendin maar morgen wil ik aleen met *name* spelen oke	Oké ik ben jouw vriendin maar morgen wil ik alleen met *Name* spelen oké
dat is beter dan een duure winkel waar ik werk in de kennel verkopen ze huisdieren daar is het heel duur *Name*	Dat is beter dan een dure winkel waar ik werk in de kennel verkopen ze huisdieren daar is het heel duur *Name*
kom je zaterdag naar moskee	Kom je zaterdag naar moskee
Daar is het niet zo gevaarlijk he.	Daar is het niet zo gevaarlijk hé.
ik moet vandaag naar de les van de boksen	Ik moet vandaag naar de les van boksen
Wat heb je gedaan vandaag?	Wat heb je gedaan vandaag?
ja ik wil dansen met u bal van zwemmen	Ja ik wil dansen met u op het bal van zwemmen
ik moet van onze juf *Name* rusten wat en voor mijn been en mijn voet doet pijn alles doet pijn	Ik moet van onze juf *Name* rusten wat en voor mijn been en mijn voet doet pijn alles doet pijn
Morgen gaan ik naar de mis en dan kerkhof met een bloem	Morgen ga ik naar de mis en dan kerkhof met een bloem
ik kom morgenvroeg met de fiets naar school en ik rijd vrijdagavond met de bus terug naar huis	Ik kom morgenvroeg met de fiets naar school en ik rijd vrijdagavond met de bus terug naar huis
video dan klik met je muis dan heb je Hotmail dan klik met je naar Hotmail	Video dan klik met je muis dan heb je Hotmail dan klik je naar Hotmail

B.2 Test Set

Original test set	**Corrected test set**
hallo hoe gaat het met jou?	Hallo hoe gaat het met jou?
wat heb je vandaag gedaan?	Wat heb je vandaag gedaan?
groetjes *name*	Groetjes *Name*
stop een keer naar mij te sturen aub	Stop een keer naar mij te sturen alstublieft
kom jij naar mijn feestje de 22ste juni of ni ik moet da wel weten dan kan ik da tegen mijn mama zeggen	Kom jij naar mijn feestje de 22ste juni of niet ik moet dat wel weten dan kan ik dat tegen mijn mama zeggen
NIKS VOOR *NAME*	Niks voor *Name*
dag *name* je hebt een mooie hond hoe heet de hond is dat geen lassie *name*	Dag *Name* je hebt een mooie hond hoe heet de hond is dat geen Lassie *Name*

dag *name* ik mag van mijn papa naar u afscheidfeestje komen *name* is dat niet goed of wel goed *name* groetjes van je beste vriendien *name* ???	Dag *Name* ik mag van mijn papa naar uw afscheidsfeestje komen *Name* is dat niet goed of wel goed *Name* groetjes van je beste vriendin *Name*???
S7	S7
dag *name* ik heb aan *name* een flesje kola lait gegeven en dan begint hij ruzie te zoeken gisteren zij *name* dat ik niks doe op de kennel dat kan niet meer *name*	Dag *Name* ik heb aan *Name* een flesje Cola Light gegeven en dan begint hij ruzie te zoeken gisteren zei *Name* dat ik niks doe op de kennel dat kan niet meer *Name*
hallo, juf *name* ik zal een berichtje sturen naar haar ik ga schrijven, ik hoop dat *name* genezen is en zij goed kan stappen, groetjes van *name*	Hallo, juf *Name* ik zal een berichtje sturen naar haar ik ga schrijven, ik hoop dat *Name* genezen is en zij goed kan stappen, groetjes van *Name*
Dag *name* hoe is het met je broertje ?	Dag *Name* hoe is het met je broertje?
computer telefoon ik hoor u wel maar gij hoort mij niet	Computer telefoon ik hoor u wel maar gij hoort mij niet
dag *name* je hebt een berichtje gestuurd maar er staat niks op	Dag *Name* je hebt een berichtje gestuurd maar er staat niks op
de kotmadam 20.10 uur	De Kotmadam 20.10 uur
hallo *name*!	Hallo *Name*!
alles goed met jou?	Alles goed met jou?
zaterdag ik opstaan wc bad tanden-poetsen boterham appelsiensap suikerklontje suiker melk	Zaterdag ik opstaan WC bad tanden poetsen boterham appelsiensap suikerklontje suiker melk
wat grapeg waneer kom je terug	Wat grappig wanneer kom je terug
ben jij verlief op *name* s10	Ben jij verliefd op *Name* s10
super goed heeeeeeel leuk ik heb lekker ge eten en hoe was u fesst van de comunnie	Super goed heel leuk ik heb lekker gegeten en hoe was uw feest van de communie
name op wie ben jij verliefde kusjes van *name* verlegen liefhebben liefde mijn-vriend is *name*	*Name* op wie ben jij verliefd kusjes van *Name* verlegen liefhebben liefde mijn vriend is *Name*
Weten jullie wat we deze namiddag gaan dioen?	Weten jullie wat we deze namiddag gaan doen?
gij zijt ook geen mozlim	Gij zijt ook geen moslim
waaarom lach je	Waarom lach je
jij moet niet liegen oke	Jij moet niet liegen oké
he ik ben niet *name* want ik ben wel *name*	Hé ik ben niet *Name* want ik ben wel *Name*
wat bedoeje *name*	Wat bedoel je *Name*
jaaaa we gaan naar elkekaar sturen	Ja we gaan naar elkaar sturen
halo juf *name* ikhoop dat zesnel trug komt groeten van *name*	Hallo juf *Name* ik hoop dat ze snel terugkomt groeten van *Name*
Dag lieve *name*, Dank je wel dat je aan mijn verjaardag denkt!!	Dag lieve *Name*, dank je wel dat je aan mijn verjaardag denkt!!
s7 Hoe gaat het met jou?	s7 Hoe gaat het met jou?
Ik ben in maart in jouw klasje komen kijken.	Ik ben in maart in jouw klasje komen kijken.

Maar je was op uitstap met de juf en de kinderen.	Maar je was op uitstap met de juf en de kinderen.
Spijtig!	Spijtig!
Ik mis je hier in De Brug!	Ik mis je hier in De Brug!
Maar misschien kom je nog eens kijken naar ons schoolfeest.	Maar misschien kom je nog eens kijken naar ons schoolfeest.
Dan kan ik je nog eens zien.	Dan kan ik je nog eens zien.
Een dikke knuffel, juf *name*	Een dikke knuffel, juf *Name*
hallo alles goet met jou en mijn ouders zus zijn supper blij en ik ben supper blij meer bewegen sporten ik ga blijven dieet soms is moelijke dieet danku juf en meester groetjes *name*	Hallo alles goed met jou en mijn ouders zus zijn superblij en ik ben superblij meer bewegen sporten en ik ga blijven diëten soms is moeilijk dieet dank u juf en meester groetjes *Name*
dag *name* ik kom maandag trhug naar school en ist kooken morgen s10	Dag *Name* ik kom maandag terug naar school en is het koken morgen s10
Hallo , *name* ik ben de neef van *name* en ik kan ook op WAI-NOT en zij heeft gezegt Dat jij zijn bveste vriendin bent s2 s4 s4	Hallo, *Name* ik ben de neef van *Name* en ik kan ook op Wai-Not en zij heeft gezegd dat jij zijn beste vriendin bent s2 s4 s4
Dat is geen goed nieuws van de poes.	Dat is geen goed nieuws van de poes.
Ocharme!	Ocharme!
Fijn trip aan de zee en breng de zon mee, ik heb ze nodig volgende zondag!	Fijne trip aan de zee en breng de zon mee, ik heb ze nodig volgende zondag!
Juf *name*	Juf *Name*
hoi *name* hoe gaat het met jou heb je al veel pijn ik wens je en veel beterschap	Hoi *Name* hoe gaat het met jou heb je al veel pijn ik wens je veel beterschap
dag *name*, bedankt dat jij mijn vriend wou zijn in wai not!	Dag *Name*, bedankt dat jij mijn vriend wou zijn in Wai-Not!
ik vind je ook heel leuk dus dank je dat jij mijn vriend wou zijn!	Ik vind je ook heel leuk dus dank je dat jij mijn vriend wou zijn!
hoeveel jaar ben jij?	Hoeveel jaar ben jij?
ik ben 13 jaar en ik verjaar op 15 mei.	Ik ben 13 jaar en ik verjaar op 15 mei.
waar woon jij?	Waar woon jij?
ik woon in berlaar.	Ik woon in Berlaar.
sorry voor het lang wachten!	Sorry voor het lange wachten!
daag.	Daag.
groetjes, *name*.	Groetjes, *Name*.
s10 s2 s7	s10 s2 s7
dag *name* ik vertrek morgen naar kokzijde ik kom zondag terug dan zal ik je een berichtje schrijven hoe het is geweesd *name* s7	Dag *Name* ik vertrek morgen naar Koksijde ik kom zondag terug dan zal ik je een berichtje schrijven hoe het is geweest *Name* s7
dag *name* ik kan vandaag niet komen naar het zwemmen groetjes van *name*	Dag *Name* ik kan vandaag niet komen naar het zwemmen groetjes van *Name*

hoi juf *name* hoe is het met jou heb je veel plastike dopje s7 ik wil deze site op wai not zetten voor dat je brieven in braille kunt schrijven de site is the name game braille bug

dag *name* ik vind het heel erg voor je waneer is dat gebeurd daarnet of daar straks maar meisje tog dat doet veel verdriet kun je ze in de tuin begraven of ga dat niet pak dan een grote zwarte zak en legt ze daar in en begraaf ze dan ik krijg trannen in mijn ogen ik vind het echt vrezelijk had de bestuurder de poes dan niet gezien godverdome wat een lope ezel is me dar ik hoop dat je een nieuwe poes krijgt ik heb er geen anders had ik jou een gegeven veel sterkte *name*

Hoe moet ik een nieuw email adres toevoegen ?

Ik heb de vorige mail per ongeluk gewist Groetjes, *name*

mijn poes is dood door domme auto

Dag meester *name* ik heb frietje gegeten het was heel lekker en er zat mayonaise bij Groetjes *name*

ja maar morgen ga dat niet ik ga op kamp en ik mochts niks mee doen van men juf

ik vint da niet leuk als jij vraagt wil jij mijn vriend worden

Hallo *name*, van wie weet je dit?

Ik weet dat de mama wel ernstig ziek is.

Ze is in het ziekenhuis, *name*

Hallo *name*, Huiswerk niet vergeten he.

Groeten juf *name*

Dag *name*, zijn je ouders al geweest?

Leuk dat ze nog op bezoek komen voor ze naar zee gaan.

Tot straks

is dat met je oude man ook het geval dat ze naar een rust huis gaat als gij niet meer voor de oude man kan zorgen of niet als jij heel droevig bent voor de oude man dan ga ik je trosten oke Groetjes *name*

dag juf *name*, ik ben in het wiekent van 17 en 19 mei bij papa en het volgende wiekent bij mama.

Hoi juf *Name* hoe is het met jou heb je veel plastieken dopjes s7 ik wil deze site op Wai-Not zetten voor dat je brieven in braille kunt schrijven de site is The Name Game Braille Bug

Dag *Name* ik vind het heel erg voor je wanneer is dat gebeurd daarnet of daarstraks maar meisje toch dat doet veel verdriet kun je ze in de tuin begraven of gaat dat niet pak dan een grote zwarte zak en leg ze daarin en begraaf ze dan ik krijg tranen in mijn ogen ik vind het echt vreselijk had de bestuurder de poes dan niet gezien godverdomme wat een lompe ezel is me dat ik hoop dat je een nieuwe poes krijgt ik heb er geen anders had ik jou een gegeven veel sterkte *Name*

Hoe moet ik een nieuw emailadres toevoegen?

Ik heb de vorige mail per ongeluk gewist Groetjes, *Name*

Mijn poes is dood door domme auto

Dag meester *Name* ik heb frietjes gegeten het was heel lekker en er zat mayonaise bij Groetjes *Name*

Ja maar morgen gaat dat niet ik ga op kamp en ik mocht niks meedoen van mijn juf

Ik vind dat niet leuk als jij vraagt wil jij mijn vriend worden

Hallo *Name*, van wie weet je dit?

Ik weet dat de mama wel ernstig ziek is.

Ze is in het ziekenhuis, *Name*

Hallo *Name*, huiswerk niet vergeten hè.

Groeten juf *Name*

Dag *Name*, zijn je ouders al geweest?

Leuk dat ze nog op bezoek komen voor ze naar zee gaan.

Tot straks

Is dat met je oude man ook het geval dat ze naar een rusthuis gaan als gij niet meer voor de oude man kan zorgen of niet als jij heel droevig bent voor de oude man dan ga ik je troosten oké Groetjes *Name*

Dag juf *Name*, ik ben in het weekend van 17 en 19 mei bij papa en het volgende weekend bij mama.

ik geef na de zomervakantie mijn wiekens niet meer door aan jou. tot straks op school! groetjes, *name*. haloo *name* dat geef niets totdan van *name* dag *name* hoe gaat het met jou alles goed ik heb een nieuwe klerekast gekregen en een burau om te knutselen groetjes *name* s7 hallo juf *name* ik ben vrerdrietig omdat er vijf pokemon van mij kwiijt zijn	Ik geef na de zomervakantie mijn weekends niet meer door aan jou. Tot straks op school! Groetjes, *Name*. Hallo *Name* dat geeft niets tot dan van *Name* Dag *Name* hoe gaat het met jou alles goed ik heb een nieuwe kleerkast gekregen en een bureau om te knutselen groetjes *Name* s7 Hallo juf *Name* ik ben verdrietig omdat er vijf Pokémon van mij kwijt zijn

Appendix C

Klare Taal Checklist

The Klare Taal (Clear Language) checklist[1]. Translations are those of the author.

C.1 The Word Level

- Korte woorden gebruikt of samengestelde woorden gesplitst?
  Did you use short words and did you split compounds?
- Enkel alledaagse en internationale woorden gebruikt?
  Did you only use everyday and international words?
- Geen vaktaal gebruikt?
  Did you avoid the use of jargon?
- Figuurlijk taalgebruik vermeden?
  Did you avoid the use of metaphorical language?
- Letterwoorden en afkortingen vermeden?
  Did you avoid the use of acronyms and abbreviations?
- Nominaliseringen vermeden?
  Did you avoid nominalisations?
- Cijfers in plaats van voluit geschreven getallen gebruikt?
  Did you use Arabic numerals instead of writing the numbers in words?
- Voorzetselketens vermeden?
  Did you avoid the use of subsequent prepositional phrases?

[1] http://www.klaretaalrendeert.be/files/Checklist%20duidelijk%20geschreven%20taal.pdf

C.2 The Sentence Level

- Spreektaal - geen schrijftaal - gebruikt?
  Did you use informal language, not formal language?
- Instructies gebruikt?
  Did you use instructions?
- Actief geschreven?
  Did you write your message using the active voice?
- Heb je korte zinnen gebruikt maar telegramstijl vermeden? (+/- 10 woorden)
  Did you use short sentences? (10 words on average)
- Staat het onderwerp voorop in de zinnen?
  Is the subject placed at the beginning of the sentence?

C.3 Text Structure

- Is de structuur logisch en eenvoudig? Een nieuwe alinea voor nieuwe informatie?
  Is the structure logical and easy and did you start a new paragraph for every new information unit?
- Duidelijke titels gebruikt?
  Did you use clear titles?
- Zijn de alinea's kort?
  Are the paragraphs short?

C.4 Design

- Is het lettertype groot genoeg?
  Is the font big enough?
- Eén lettertype gebruikt?
  Did you use one font?
- Hoofdletters vermeden?
  Did you avoid capital letters?

- Vet gebruikt om de nadruk te leggen? Niet onderstreept?
  Did you use boldface to highlight words instead of underlining them?
- Duidelijke, functionele illustraties of pictogrammen gebruikt?
  Did you use clear illustrations or pictographs?
- Voldoende ruimte tussen de alinea's?
  Is there enough space between the paragraphs?
- Data voluit geschreven?
  Did you write dates in full?

APPENDIX **D**

Pseudocode for the Syntactic Simplification Module

D.1 Creating Sentences

```
Open the syntactically annotated sentences generated by Alpino.
Search for main clauses (SMAIN, DP, or SV1) in the syntactically annotated sentences.
For each main clause:
    If the main clause contains embedded clauses (subordinate, relative, participial phrases,
    (om) te + infinitive, appositives):
        Remove the embedded clauses from the main clause.
        Check for more clauses within the embedded clauses recursively.
        For each (non-main) clause:
            Create a sentence (a hierarchical structure of phrases and words with features,
            taken from the original clause).
            If the clause contains a passive construction:
                Mark the sentence as passive.
            End if
            If the clause is a subordinate clause:
                Remove the subordinate conjunction.
                If the clause is headed by a question word:
                    Mark the sentence as interrogative.
                End if
                If the clause expresses temporality or concession:
                    Depending on the conjunction type, move the sentence in front
                    of or behind the main clause.
                End if
```

```
            Elsif the clause is a participial phrase or an (om) te + infinitive clause:
                Identify the subject in the syntactically annotated sentences.
                If the subject is a noun phrase:
                    Add the head noun as a feature.
                Else:
                    Add the subject as a feature.
                End if
            Elsif the clause is a relative clause:
                Identify the antecedent and its grammatical function in the syntactically
                annotated sentences.
                If the antecedent is a noun phrase:
                    Add the head noun and its function as a feature.
                Else:
                    Add the antecedent and its function as a feature.
                End if
            Elsif the clause is an appositive:
                Identify the subject in the syntactically annotated sentences.
                If the subject is a noun phrase:
                    Add the head noun as a feature.
                Else:
                    Add the subject as a feature.
                End if
            End if
        End for
    End if
    Create a sentence (a hierarchical structure of phrases and words with features, taken
    from the original main clause).
    If the main clause is SV1 or is headed by a question word:
        Mark the sentence as interrogative.
    End if
    If the main clause is missing an overt subject as a result of ellipsis:
        Identify the subject in the syntactically annotated sentences.
        If the subject is a noun phrase:
            Add the head noun as a feature.
        Else:
            Add the subject as a feature.
        End if
    End if
    If the main clause contains a passive construction:
        Mark the sentence as passive.
    End if
End for
```

D.2 Changing the Order of the Constituents

```
For each newly created sentence:
    If a prep. phrase or an adverb(ial phrase) is located at the beginning of the sentence:
        Move it to the back of the sentence (i.e., detopicalise it).
    End if
    If a word with a negative polarity is found:
        Look for the head word it modifies and attach it to the head.
    End if
    Find all verbs in the sentence, cluster them, and create a verb group.
    If the original clause was a main clause, subordinate clause, or (om) te + inf. clause:
        If the subject was covert:
            Place the retrieved antecedent at the beginning of the sentence.
            Place the verb group behind the antecedent.
        Else:
            Place the verb group behind the subject.
        End if
        If the sentence carries the interrogative feature:
            Place the question word at the beginning of the sentence.
        End if
    Elsif the original clause was a part. phrase or a rel. clause with a subject antecedent:
        Place the verb group at the beginning of the sentence.
        Place the retrieved antecedent in front of the verb group.
    Elsif the original clause was a relative clause with an (in)direct object antecedent:
        Place the verb group behind the subject.
        Place the retrieved antecedent behind the verb group.
    Elsif the original clause was an appositive:
        Place to be at the beginning of the sentence.
        Place the retrieved antecedent in front of to be.
    End if
    If the sentence carries the passive feature:
        Check for a prepositional phrase with an agent.
        If an agent is found:
            Switch the positions of the agent and the patient.
            Remove worden 'to be'.
        End if
    End if
End for
```

APPENDIX E

Description of Survey Participants

This part is dedicated to the user survey on syntactic simplification. Appendix E.1 gives an overview of the age, daily environment, and communicative skills of the target users. In Appendix E.2, we discuss the target users' social media use. Appendix E.3 is an overview of the comments that were made with respect to the syntactic difficulty of individual pictograph sequences.

E.1 Target Users' Age, Daily Environment, and Communicative Skills

#	Relation to user	User's age	User's daily environment	User's expressive skills	User's receptive skills	User's pictograph experience
#1	Teacher	15	Special education	Uses language to communicate	Understands language	No experience
#2	Family	60	Residential group	Uses pictographs to communicate	Understands language	Offline use
#3	Speech therapist	15	Special education	Uses language to communicate	Understands language	Offline & online use
#4	Speech therapist	18	Special education	Uses language to communicate, but can't type or write	Understands language, but needs pictograph support for rules	Offline & online use
#5	Teacher	12	Special education	Severe expressive limitations	Understands language	Offline & online use
#6	Teacher	11	Special education	Uses pictographs to communicate	Understands language	Offline & online use
#7	Teacher	13	Special education	Uses pictographs and some language (echolalia) to communicate, but only communicates in familiar and safe settings	Understands language in familiar and safe settings, but needs pictograph support for daily schedules	Offline use
#8	Teacher	8	Special education	Uses language and gestural communication	Needs gestures and pictographs to understand language	Offline use
#9	Paramedic	16	Special education	Uses language to communicate, but often without real content, and has trouble coming up with words	Understands language partly, and sometimes does not answer questions correctly	Offline use
#10	Carer	56	Day centre	Uses pictographs and gestural communication	Understands language when gestures are used	Offline use
#11	Teacher	9	Special education	Uses pictographs to communicate	Understands language	Offline use

#12	Family	34	Day centre	Uses pictographs to communicate	Understands language, but can't read	Offline & online use
#13	Speech therapist	6-12	Special education (multiple children with ID)	Use pictographs to communicate	Severe receptive limitations	Offline & online use
#14	Teacher	12-24	Special education (multiple teenagers with ID)	Use pictographs to communicate	Understand language	Offline use
#15	Carer	30	Residential group	Uses language to communicate	Understands language	Offline use
#16	Carer	52	Residential group	Uses (simple) language to communicate	Understands language	Offline use
#17	Teacher	15	Special education	Uses language to communicate	Understands language	Offline use
#18	Family	15	Special education	Severe expressive limitations	Understands language	Offline & online use
#19	Carer	30	Residential group	Uses language to communicate	Understands language	Offline use
#20	Teacher	10	Special education	Uses pictographs to communicate	Understands language when gestures are used	Offline use
#21	Carer	30	Day centre	Uses pictographs to communicate	Understands language, but would probably benefit from pictograph support	Offline & online use
#22	Family	6	Elementary school	Uses language to communicate, except when upset	Understands language, but needs pictograph support for daily schedules	Offline use
#23	Carer	13-22	Special education (multiple teenagers who are deaf and have ID)	Most teenagers use signs and pictographs to communicate	Understand sign language, but need pictograph support for daily schedules, social situations, or new signs	Offline use

#24	Carer	35	Day centre	Uses pictographs to communicate	Needs pictographs to understand language	Offline use
#25	Carer	18	Special education	Uses language to communicate, but has trouble expressing herself online	Understands language when slowly spoken	Offline & online use
#26	Carer	52	Residential group	Uses language to communicate	Understands (simple) language	Offline use
#27	Speech therapist	28	Residential group	Uses (simple) language to communicate	Understands language	Offline use
#28	Teacher	8	Elementary school	Uses language to communicate	Understands language, but does not understand emotions	Offline use

E.2 Target Users' Social Media Use

#	**Devices used**	**Social media use**	**Why does the user use social media?**	**What makes social media difficult to use?**
#1	Computer, smartphone, tablet	Daily	Interacting with others, sharing pictures, posting messages	Not knowing how to behave in an appropriate way, navigation
#2	Computer	Monthly	Watching pictures of family and friends, answering messages with the help of a carer	The user requires personal assistance to communicate online
#3	Computer, tablet	Rarely	The user does not (yet) use social media websites	Social media websites are difficult to use in general, privacy issues
#4	Smartphone	Daily	Communicating with parents and teachers through WhatsApp	Reading and writing difficulties, lack of privacy, lack of personal assistance
#5	Computer, tablet	Rarely	Playing games	Reading difficulties

#6	Computer, tablet	Rarely	Communicating with family and friends	Reading difficulties
#7	Computer, tablet	Never	The user does not (yet) use social media websites	Social media websites are difficult to use in general, privacy issues
#8	Tablet	Never	The user does not (yet) use social media websites	Social media websites are difficult to use in general
#9	Computer, tablet	Never	The user does not (yet) use social media websites	Social media websites are difficult to use in general
#10	Tablet	Never	The user does not (yet) use social media websites	Social media websites are difficult to use in general
#11	Tablet	Never	The user does not (yet) use social media websites	The user's expressive skills are limited
#12	Computer, smartphone, tablet	Weekly	Communicating with friends, watching YouTube videos, boosting his/her self-esteem	The interfaces are difficult to use
#13	None	Never	The users do not (yet) use social media websites	Social media websites are difficult to use in general
#14	Computer, tablet, hand-made booklets	Never	The users do not (yet) use social media websites	Reading and writing difficulties
#15	None	Never	The user does not (yet) use social media websites	Social media websites do not interest the user
#16	None	Never	The user does not (yet) use social media websites	Reading and writing difficulties
#17	Nintendo	Rarely	Watching YouTube videos and music clips, boosting his/her self-esteem	Reading difficulties
#18	Computer, smartphone, tablet	Weekly	Playing games and watching YouTube videos, communicating with his/her parents about practical things, such as estimated time of arrival	Being misunderstood by other people
#19	None	Never	The user does not (yet) use social media websites	Social media websites are difficult to use in general
#20	None	Never	The user does not (yet) use social media websites	The user is still too young to use social media websites

#21	Computer	Never	The user does not (yet) use social media websites, but the carer remarks that specialised communication aids could be helpful. The user likes to watch pictures. Personal assistance is needed to type messages on the computer	There is no (or very limited) access to computers in the day centre
#22	None	Never	The user does not (yet) use social media websites	The user is still too young to use social media websites
#23	Computer, smartphone, tablet, Nintendo	Daily	Communicating with friends and family, staying in touch with people, having a good time	Reading and writing difficulties, being misunderstood because of writing difficulties, which leads to frustration. Many of the users must be assisted or controlled (which is a time-consuming task for the carers)
#24	Computer	Rarely	The user does not (yet) use social media websites	Too much information for the user to filter
#25	Computer, smartphone, tablet	Daily	Communicating with friends and other people in the user's environment	Reading and writing difficulties, difficulties understanding the content of private messages, privacy issues (such as adding strangers)
#26	None	Never	The user does not (yet) use social media websites, but the carer remarks that social media could allow the user to stay in touch with his/her friends	Reading and writing difficulties
#27	Computer	Weekly	Sending emails	Too much information for the user to filter
#28	Computer, tablet	Rarely	The user does not (yet) use social media websites	Social media websites are difficult to use in general

E.3 Comments Made with Respect to Syntactic Difficulty

An overview of all comments that were left by the participants of our survey with respect to syntactic difficulty. Note that comments were optional and left empty in most cases. Translations are those of the author.

Sentence type	#	Comment	Translation
Passive non-simplified	9	"zinsbouw moeilijk te interpreteren als actief of passief."	"Syntax difficult to interpret as active or passive."
	17	"Niet duidelijk over wie het gaat"	"Unclear who this is about."
	20	"onduidelijk, is hij aangereden of reed hij zelf met de brommer?"	"Unclear, was he hit or was he driving a motorcyle himself?"
	21	"Is hij aangereden of heeft hij iemand aangereden?"	"Was he hit or did he hit someone?"
	26	"te weinig info. Wie is aangereden?"	"Not enough information. Who is being hit?"
Passive simplified	21	"verwarring of hij pest, of gepest wordt"	"Confusion whether he does the bullying, or is being bullied."
	28	"Is het het meisje dat pest of "ik" Indien het het meisje is, is het woord "ik" erbij verwarrend. Indien het "ik" is, zou dit misschien beter vooraan staan."	"Is the girl the one who does the bullying or "I"? If it is the girl, the "I" is confusing. If it is "I", it might be better to put that one in the front."
Relative clause non-simplified	3	"langere zin, maar wel haalbaar"	"Longer sentence, but feasible."
	4	"te lange zin"	"Sentence is too long."
	5	"te lang"	"Too long."
	6	"lange zin en betekenis niet duidelijk door taalbeperking"	"Long sentence and meaning not clear due to language disability."
	8	"teveel picto's (te ingewikkelde zin)"	"Too many pictographs (sentence is too complex)"
	12	"de opbouw van de zin is te moeilijk - ik zou hem als volgt opbouwen [jij] [sturen] [mij] [bericht], [bericht] [zijn] [goed]"	"The syntax of the sentence is too difficult - I would structure it as follows: [you] [send] [me] [message], [message] [are] [good]."
	13	"Te lang"	"Too long."
	15	"Te lang "	"Too long."
	18	"de veel details"	"Too many details."
	20	"veel te lange zin."	"Sentence is much too long."
	23	"te lange zin"	"Sentence is too long."

	25	"zin te lang om samenhang te zien"	"Sentence is too long for it to be coherent."
Relative clause simplified	3	"heel duidelijk "	"Very clear."
	9	"ok"	"OK."
	19	"zou misschien kunnen lukken"	"Should be feasible."
	20	"dit is duidelijker"	"This is clearer."
Subordination non-simplified	5	"dubbele boodschap"	"The message is twofold."
	8	"enkel 2de deel is nuttig/begrijpbaar eerste deel is te beperkt"	"Only the second part is useful/understandable. The first part is too restricted."
	11	"te moeilijke constructie"	"Construction is too difficult."
	13	"Opnieuw te lange zin"	"Again, sentence is too long."
	23	"te complexe zin"	"Sentence is too complex."
	25	"te lang ze gebruikt vaak 1 picto om iets te gaan zeggen"	"Too long. She usually uses one pictograph to say something."
Subordination simplified	13	"Vrij duidelijk, korte boodschappen"	"Rather clear, short messages."
	23	"duidelijkere boodschap"	"Message is clearer."
	26	"picto's zijn duidelijk."	"Pictographs are clear."
Coordination non-simplified	3	"lange zin, maar wel heel duidelijk"	"Long sentence, but very clear."
	4	"te lang meerdere korte zinnen van maken"	"Too long, make multiple shorter sentences from it."
	5	"te lang"	"Too long."
	6	"te veel taal, onnodige info. Enkel belangrijkste zou moeten weergegeven worden. te letterlijk"	"Too much language, useless information. Only the most essential parts should be shown. Too literal."
	8	"teveel picto's, te lange zin"	"Too many pictographs. Sentence is too long."
	10	"teveel prenten voor 1 zin *Name* zou er maar een aantal uithalen om te vertellen"	"Too many images for one sentence. *Name* would only use a few of them to communicate."
	11	"de lange zin opdelen in stukken zou makkelijker zijn"	"It would be easier to split the long sentence into several parts."
	12	"zin te lang - beter nieuwe zin starten. Hem verwijst naar de baby. Dat wordt moeilijk. Beter de baby nog eens herhalen."	"Sentence is too long - better start a new sentence. "Him" refers to the baby. That will be difficult. Better repeat the baby."
	13	"veel te lang"	"Much too long."
	17	"Meer pictogrammen en tekens dan de andere"	"More pictographs and symbols than the other ones."
	19	"te ingewikkeld"	"Too complex."

	20	“Er zitten overbodige pictogrammen tussen.”	“There are redundant pictographs.”
	22	“ik vind persoonlijk dat er teveel picto’s worden gebruikt... voor elk woord? zijn minder picto’s niet duidelijker, ...”	“I personally think that too many pictographs are used. For every word? Aren’t less pictographs clearer?”
	25	“te lang eerder 4 lijnen onder elkaar woensdag baby + bezoek ik + flesje geven leuk”	“Too long. Rather put four lines underneath each other. Baby + visit I + give bottle + nice.”
Coordination simplified	9	“te verwarrend in chronologie”	“Too complex in chronology.”
	14	“De zinnen zijn technisch leesbaar, maar ik snap de verschillende onderwerpen uit de zinnen niet”	“The sentences are technically readable, but I don’t understand the different topics in the sentences.”
	19	“niet spelen kent hij wel en stout ook maar of de link duidelijk zal zijn?”	“He knows “not playing” and “nasty” as well, but not sure if the link will be clear?”
	22	“de betekenis zal misschien wel begrepen worden, maar het oorzakelijk verband? waar is de ’omdat’? niet spelen omdat je niet flink was?”	“The meaning might be understood, but the causal relationship? Where is the “because”? Not playing because you were naughty?”
Appositive clause non-simplified	1	“Volgorde van de pictogrammen.”	“Order of the pictographs.”
	3	“duidelijk maar ik zou ’nieuwe’ voor ’cars’ plaatsen”	“Clear, but I would put “new” in front of “Cars”.”
	5	“rare volgorde”	“Weird order.”
	6	“onlogische volgorde gezien de betekenis”	“Illogical order given the meaning.”
	26	“te moeilijk om te begrijpen.”	“Difficult to understand.”
	28	“Volgorde van de picto’s”	“Order of the pictographs.”
Appositive clause simplified		(No comments)	(No comments)
Subordination + passivity non-simplified	4	“vind ik zelf wat moeilijk :) niet duidelijk om wie het gaat”	“I personally find this quite difficult. :) Not sure who this is about.”
	7	“moeilijk!”	“Difficult!”
	9	“De zin is voor mij ook helemaal niet duidelijk. Bovendien is het moeilijk te weten in welke volgorde de picto’s in de zinsbouw passen.”	“The sentence is not clear to me either. Even more, it is difficult to know in which order the pictographs fit into the syntax.”
	10	“te moeilijk”	“Too difficult.”
	12	“te complexe zin”	“Sentence is too complex.”
	13	“Niet duidelijk”	“Not clear.”

	14	"Moeilijker zonder context"	"More difficult without context."
	15	"Te ingewikkeld"	"Too complicated."
	19	"daar komt niks van"	"This won't work out."
	22	"ik vind de structuur van de zin niet overeen komen met de volgorde van de picto's"	"I think the structure of the sentence does not match the order of the pictographs."
	25	"onduidelijk"	"Unclear."
Subordination + passivity simplified	16	"zij zou de zinsconstructie niet kunnen achterhalen denk ik"	"She would not be able to understand the syntax, I think."
	28	"volgorde picto's"	"Order of the pictographs."

APPENDIX F

Hierarchical Structure of the Pictograph Interface

First level	Second level	Third level	Also links to
Main	**Conversation**	Hello & goodbye	
		General	
		Opinion	Positive thoughts
			Negative thoughts
		Communication	Technology
		Letters	
		Punctuation	
	Feelings & behaviour	Positive thoughts	
		Negative thoughts	
		Colours	
		Properties	
		Senses	
		Love	
		Sexuality	
		Religion	
		Behaviour	
		Festivities	
		Medical issues	Medication & aids
		Disasters & accidents	Medication & aids
	Dimensions	Weekdays, months & seasons	
		Time & clock	
		Numbers	
		Location	
	People	Pronouns	

	Family & friends	Children & babies
	Children & babies	Toys
	Professions	
	Famous characters & fantasy	
	Body	Care
		Medical issues
Animals	Dog	
	Cat	
	Rodents	
	Birds	
	Aquatic animals	
	Farm animals	
	Insects	
	Wild animals	
Leisure	Ball sports	
	Gymnastics & athletics	
	Water sports	
	Music	
	Creative	
	Free time	Toys
Locations	School	School materials
	Day centre	
	Village & city	
	Shops	Objects
	Living room	
	Kitchen	Food & drinks
		Cookware
	Bathroom	Care
	Bedroom	
	Garden	Flowers & plants
		Working materials
Clothing	Daytime clothing	
	Underwear	
	Shoes	
	Swimming clothes	
	Accessories	
Nature	Weather	
	Flowers & plants	
	Landscape	
Food & drinks	Drinks	
	Vegetables	
	Fruits	

	Meat & fish	
	Breakfast	Fruits
	Snacks	Fruits
	Warm meal	Vegetables
		Meat & fish
	Cooking & baking	Cookware
Objects	Technology	
	School materials	
	Furniture	
	Household materials	
	Working materials	
	Toys	
	Care products	
	Medication & aids	
	Cookware	
Traffic & vehicles	Transport	
	Traffic	
	Movements	

APPENDIX **G**

Overview of the Object-Oriented Framework

In this appendix, we give an overview of the object-oriented representation framework. We give a short description of the features and methods that are applicable to each of the object classes.

G.1 Features

Figures G.1 to G.4 show the objects that are used within each of the stand-alone technologies described in the previous chapters and their relevant features.

Figure G.1 presents a fully annotated message object in the final step of the Text-to-Pictograph translation process (with syntactic simplification and word sense disambiguation). A message object consists of one or multiple sentence objects, which, in turn, contain one or multiple word objects (or phrase objects that include word objects, when the syntactic simplification module is activated). Word objects are linked to lexunit objects, which link to synset objects. The pictographs, which are retrieved via their connected synsets in the WordNet database, are added as features to their corresponding word objects. Once all pictographs are retrieved, a pictopath object is created as a sentence feature to keep track of the word objects (and their associated pictographs) that need to be processed during optimal path calculation. Finally, image objects are created for the winning pictographs. The different features per object type are shown in the figure. Most of these features were already present in the baseline system, with the exception of the simplification features ({amountofverbs}, {func-

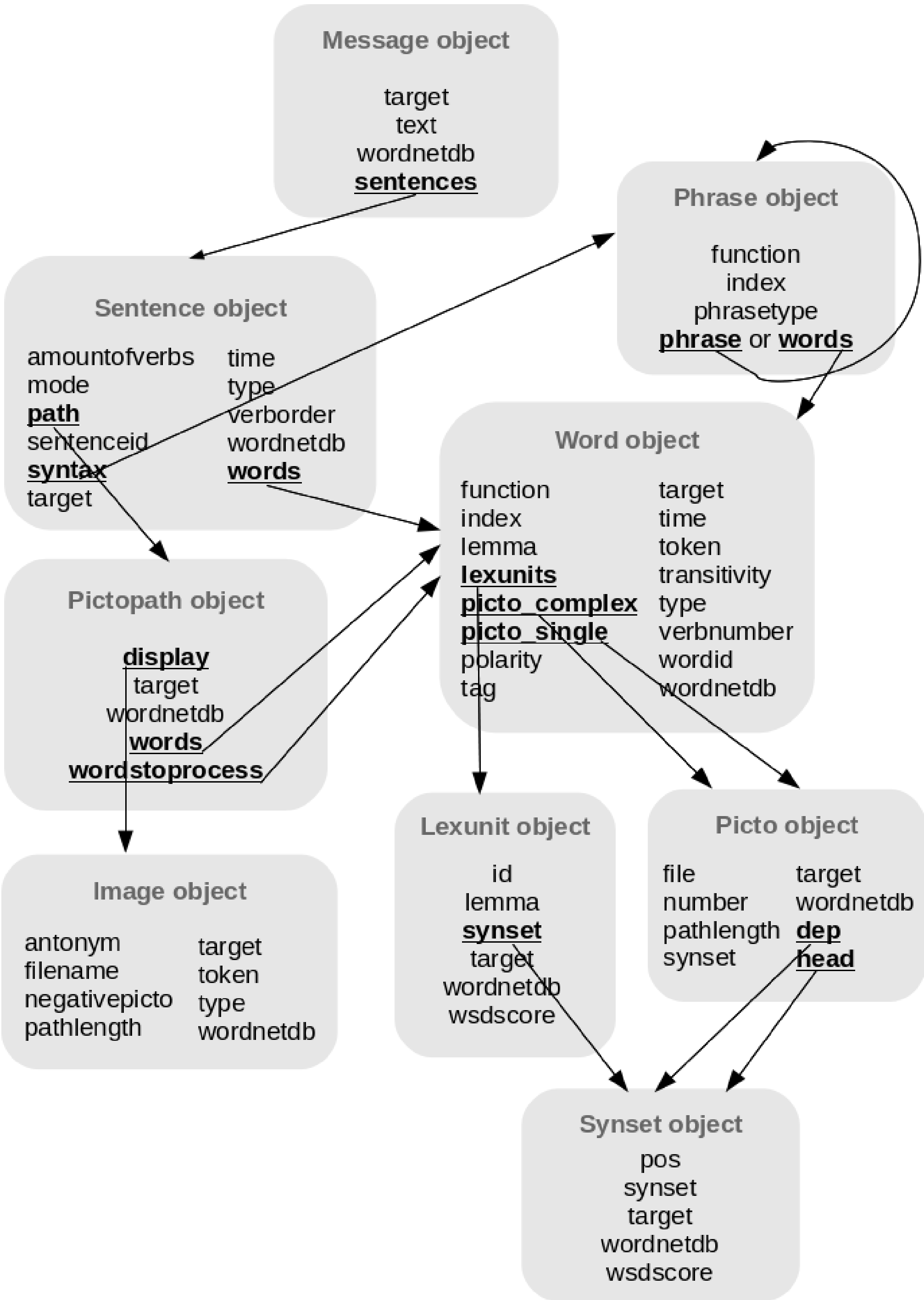


Figure G.1: Features used per object type in the Text-to-Pictograph translation tool (including simplification and word sense disambiguation).

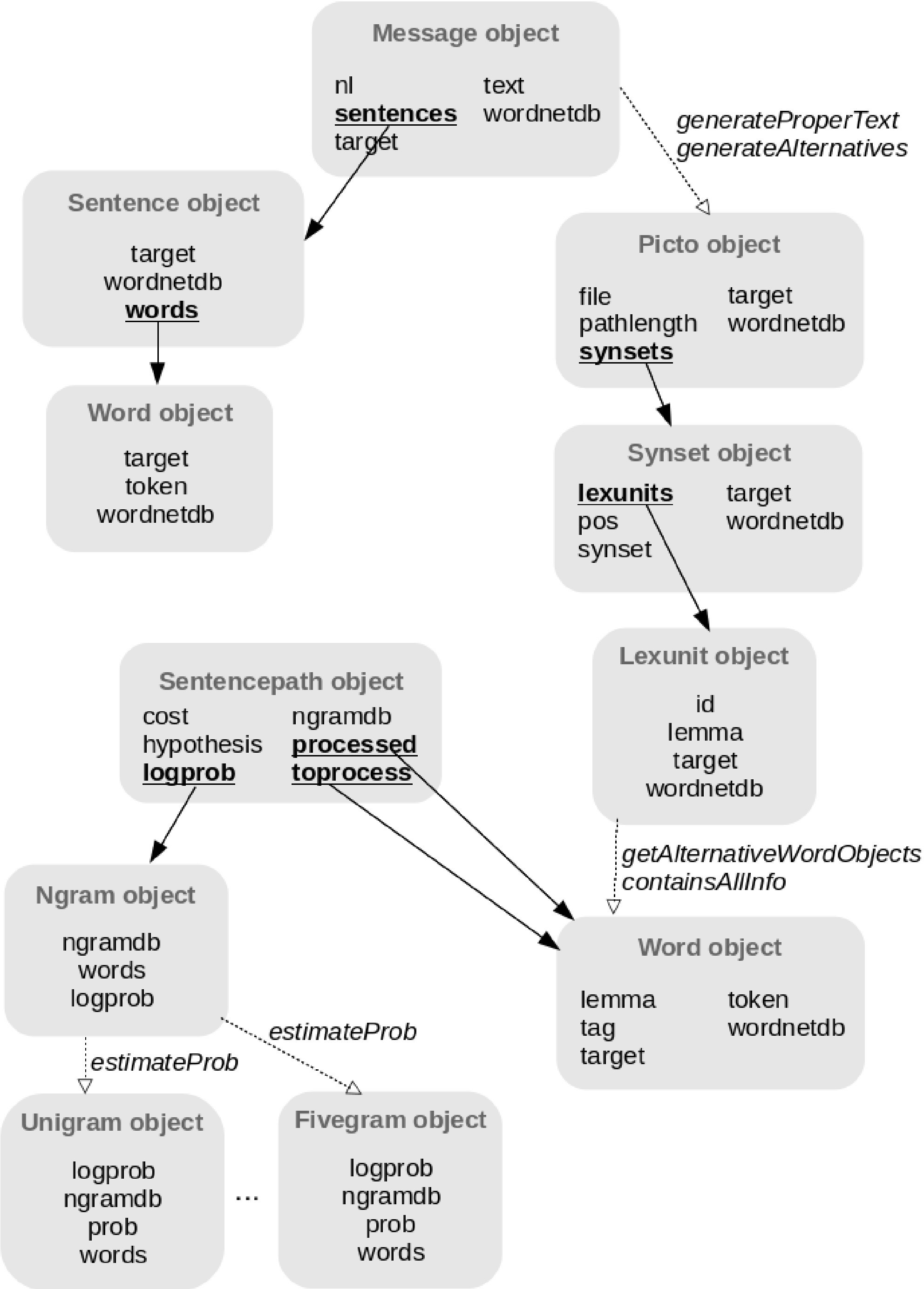


Figure G.2: Features used per object type in the Pictograph-to-Text translation tool (5-gram language modelling approach).

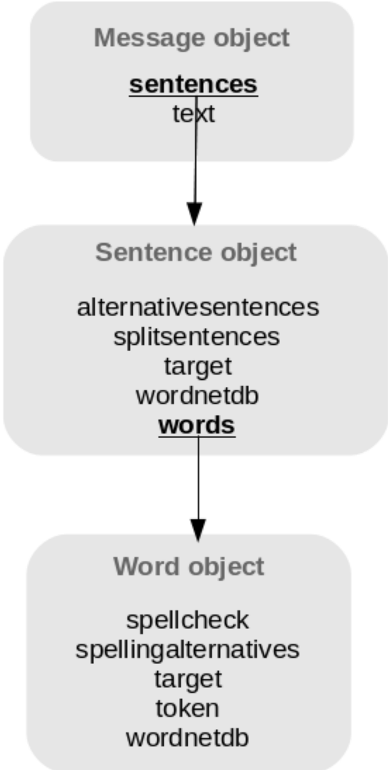


Figure G.3: Features used per object type in the spelling correction tool.

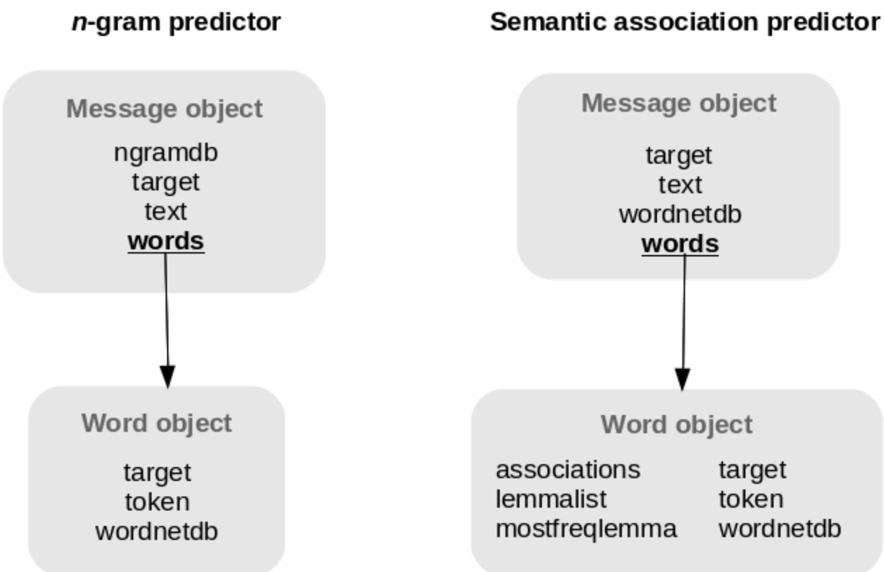


Figure G.4: Features used per object type in the prediction tools.

tion}, {index}, {mode}, {phrasetype}, {time}, {transitivity}, {type}, {verborder}) and the word sense disambiguation features ({sentenceid}, {wsdscore}, {wordid}). The phrase object is a new type of object that operates as an intermediate level between the sentence object and the word objects.

Figure G.2 shows the objects that are used in the 5-gram language modelling approach toward Pictograph-to-Text translation. Note that the machine translation-based approaches toward Pictograph-to-Text translation do not make use of the object-oriented framework. In the first step, a message object is created, containing sentence objects with pictographs, represented as individual word objects. After applying the *generateProperText* and *generateAlternatives* methods, a pictograph object is created, which link to synset objects. The synset objects, in turn, link to lexunit objects, which contain the lemmas that are used to generate the surface forms. The *getAlternative-WordObjects* and *containsAllInfo* methods lead to the creation of word objects, which are used by the sentencepath object to generate the most likely natural language string. In choosing the most probable surface forms, the sentencepath is guided by the ngram objects (unigram, bigram, trigram, fourgram, and fivegram objects), which use the *estimateProb* method to look up token sequences in an *n*-gram database.

The spelling correction tool is a stand-alone tool that uses a simple object structure to store information. The features of a fully annotated message object are shown in Figure G.3. An incoming message consists of one or multiple sentence objects, which, in turn, contain one or multiple word objects. Spelling variants for the individual words are stored in the {spellingalternatives} features, while all the combinations of variants (i.e., the hypotheses) and their corresponding split forms are stored in the {alternativesentences} and {splitsentences} features on the sentence level.

Finally, Figure G.4 shows the objects that are used in the prediction tools. Both tools create a message object with word objects, corresponding to the pictographs that were previously entered by the user. As compared to the previously discussed technologies, the semantic association predictor introduces three new features on the word level ({associations}, {lemmalist}, and {mostfreqlemma}).

G.2 Methods

G.2.1 The Object Class

new: Creates a new object of the appropriate subclass, with the appropriate features and values.

openNGramDatabase: Opens the *n*-gram database. Creates a new DBI::db object with parameters that allow connection with the PostgreSQL database containing the language-specific language model. Uses DBI::db::new.

openWordnet: Opens the WordNet database. Creates a new DBI::db object with parameters that allow connection with the PostgreSQL database containing the language-specific WordNet. Uses DBI::db::new.

pushFeature: Pushes a value into the value array of a feature. Creates the value array of the feature if it does not exist.

G.2.2 The Message Class

ISA("object")

addFullStop: Adds a full stop to the end of the message if not ending in a punctuation mark.

addPictoPaths: Adds the pictopaths that represent the message. Uses message::addPictos and message::searchArgMax.

addPictos: Adds the pictos to a message. Uses sentence::addPictos.

addSynsets: Adds synsets to the message. Uses object::openWordnet and sentence::addSynsets.

addSyntaxInfo: Step one of the simplification process. Uses Alpino's parse to build a hierarchy of sentence and word objects. Uses message::buildInterrogativeSentence, message::buildMainSentenceWithSubjectCheck, message::buildMainSentence, message::buildRelSentence, message::buildOTISentence, message::buildSSUBSentence, message::buildAppSentence, and message::buildPPRESSentence.

addWsdScores: Adds the WSD scores to their matching synsets, using the sentence IDs and word IDs to disambiguate between identical words.

analyzeTime: The main module for temporal analysis. Uses sentence::addTimeRules, sentence::changeVerbOrder, and sentence::generateTimePicto.

applyAlpino: Executes an Alpino parse. Writes the output to a treebank.

buildAppSentence: Creates a sentence object from an appositive clause. Uses message::fetchAppositionAntecedent and message::buildWordObjects.

buildInterrogativeSentence: Creates a sentence object from an interrogative clause. Uses message::fetchQuestionWord and message::buildWordObjects.

buildMainSentence: Creates a sentence object from a main clause. Uses message::buildWordObjects.

buildMainSentenceWithSubjectCheck: Creates a sentence object from a main clause that potentially has a covert subject. Uses message::buildWordObjects.

buildOTISentence: Creates a sentence object from an OTI clause. Uses message::buildWordObjects.

buildPPRESSentence: Creates a sentence object from a PPRES clause. Uses message::buildWordObjects.

buildRelSentence: Creates a sentence object from a relative clause. Uses message::fetchAntecedent and message::buildWordObjects.

buildSSUBSentence: Creates a sentence object from a subordinate clause. Uses message::buildWordObjects.

buildPhraseObjects: Creates phrase objects. Uses message::buildPhraseObjects recursively.

buildWordObjects: Creates word objects. If a node is non-terminal, it is a phrase. In that case, message::buildPhraseObjects is called.

checkCompress: Checks whether the compression module has been activated. If this is the case, sentence::compress is called.

convertWsd: Converts the WSD output into a format that can be used in the Picto tools.

CornettoWsd: The main module for WSD. Uses message::createNewTextAndLabelObjects, message::generateInputFile, message::useWsdTool, message::convertWsd, and message::addWsdScores.

createInputFileForAlpino: Creates a temporary input file for Alpino (one sentence per line).

createNewTextAndLabelObjects: Adds unique sentence IDs and word IDs to every sentence object and word object. This is necessary for disambiguation purposes.

detectSentences: Embeds the word objects into sentence objects. Uses message::tokenize if called by an untokenised message.

fetchAntecedent: Retrieves the antecedent from the Alpino parse.

fetchAppositionAntecedent: Retrieves the apposition's antecedent from the Alpino parse.

fetchQuestionWord: Retrieves the question word from the Alpino parse.

findBestBiGrams: Looks up the most probable picto bigram in the picto *n*-gram database, using the previously inserted picto as the first unigram of the bigram.

findBestTriGrams: Looks up the most probable picto trigram in the picto *n*-gram database, using the two previously inserted pictos as the first bigram of the trigram.

findSeparableVerbs: Activated when Alpino is disabled or fails. Detects separable verbs in each of the sentences of the calling message. Uses sentence::findSeparableVerbs.

generateProperText: Turns the message from Beta/Sclera into proper natural language text using the language model. Uses sentence::generateProperText.

generateStartOfSentence: Generates a beginning of sentence word object.

generateWsdInputFile: Generates an input file for the WSD tool.

HTMLOut: Provides HTML output of the message, pointing to picto images. Uses sentence::HTMLOut.

intersectAssociations: Intersects the association lists that were created by looking up every picto's most frequent lemma in the semantic space. Uses DBI::db::lookup.

JSONOut: Provides JSON output of the message. Uses sentence::JSONOut.

lemmatize: Lemmatises each of the sentences in the message using sentence::lemmatize.

lookupPictoDictionary: Looks up the words in the message in the picto dictionary. Uses object::openWordnet and sentence::lookupPictoDictionary.

openNGramDatabase: Opens the picto *n*-gram database. Creates a new DBI::db object with parameters that allow connection with the PostgreSQL database containing the language-specific language models. Uses DBI::db::new.

ParallelJSONOut: Provides parallel text and JSON output of the message. Uses sentence::ParallelOutOtherFile and sentence::ParallelJSONOut.

predictNextWord: Predicts the next picto using picto *n*-grams. Uses message::openNGramDatabase, message::generateStartOfSentence, message::findBestBiGrams, and message::findBestTriGrams.

preProcess: Removes emoji.

retrieveAssociations: Retrieves associated pictos for pictos that are previously inserted by the user. Uses object::openWordnet, word::retrieveLemmas, word::findMostFrequentLemma, word::findAssociations, word::lemmatizeAssociations, word::findMostFrequentDepLemma, word::findDepAssociations, and word::lemmatizeDepAssociations.

searchArgMax: Looks for the optimal path to convert the message into pictos. Uses sentence::searchArgMax.

simplify: Step two of the simplification process. Re-orders the syntactic constituents. Uses sentence::movePPsToBack, sentence::checkForPassives, sentence::addPolarity, sentence::changeOrder, sentence::buildWordObjects, and sentence::addQuestionMark.

spellCheck: Spell checks and corrects the input message. Uses sentence::removeCapitalFirstWord, sentence::changeAllUpperCaseToLower, sentence::convertContractionsAndCommonAbbreviations, sentence::findNonWords, sentence::findVariants, sentence::filterVariants, sentence::buildAllSentences, sentence::makeSplitSentences, and sentence::fuzzyMatch.

tag: Activated when Alpino is disabled or fails. Tags the calling message. Writes a temporary output file to timestamp. Uses the Hunpos tagger and writes the output of the tagger to timestamp.tmp. Reads in timestamp.tmp, and creates a new word object per line in timestamp.tmp.

taglemmatize: Either activates the sub-modules of the original baseline taglemmatize module, or executes an Alpino parse and applies a series of simplification operations. Uses message::addFullStop, message::preProcess, message::createInputFileForAlpino, message::tokenize, message::tag, message::detectSentences, message::findSeparableVerbs, message::lemmatize, message::applyAlpino, message::addSyntaxInfo, message::simplify, message::analyzeTime, and message::checkCompress.

TextOut: Provides textual output of the message. Uses sentence::TextOut.

tokenize: Tokenises the calling message. Turns a text string into a list of word objects.

useWsdTool: Activates the WSD tool. Writes the output to a temporary file.

WordBuilder: Checks for every two tokens whether they are non-words or infrequent real words. If this is the case, checks if their compound exists (with a minimum frequency).

G.2.3 The Sentence Class

ISA("object")

adaptPolarity: Activated when Alpino is disabled or fails. If a negative word is found, looks for the head of this word and puts the negative word in the {polarity} feature of the head word. Removes the negative word from the word list.

addObjectBehindVerbs: Adds the object antecedent behind the verb group.

addPictos: Adds pictos to the words in the sentence. Uses word::addPictosNotInWordnet, word::lookupPictoDictionary, word::isContentWord, and word::addPictos.

addPolarity: Gives words a {polarity} feature if a negative is found.

addSubjectToFront: Adds the subject antecedent to the beginning of the sentence.

addSynsets: Adds synsets to the message. Calls object::openWordnet and word::addWordNet.

addTimeRules: Counts the amount of verbs in the verb group. Activates the temporal analysis rules. Uses sentence::applyVerbRules.

addToBeInFront: Creates a "to be" word object and adds it to the beginning of the sentence.

addQuestionMark: Adds a question mark word object to the end of the sentence if the sentence type is interrogative.

addQuestionWordToFront: Adds the question word to the beginning of the sentence.

applyVerbRules: Applies the rules for verb group simplification. Marks temporality and verb order.

buildAllSentences: Creates all possible combinations of spelling variants, resulting in a number of hypotheses, which are pushed into the {alternativesentences} feature.

buildWordObjects: Flattens the word object structure after syntactic simplification took place. Uses sentence::getWords.

changeAllUpperCaseToLower: Converts sentences that are written in all uppercase characters into lowercase characters.

changeOrder: Re-orders verbs, subjects, and antecedents. Uses sentence::findAllVerbs, sentence::moveMainAndSSUBVerbs, sentence::moveRelObjectVerbs, sentence::moveRelSubjectVerbs, sentence::createAppositiveSentences, sentence::moveQuestionVerbs, and sentence::makeActive.

changeVerbOrder: Changes the order of the verbs in the verb group. Deletes verbs that do no longer contribute to the overall meaning of the verb group.

checkForPassives: Checks if the sentence is a passive sentence and adds this as a sentence-level feature. Uses sentence::checkPhraseForPassives.

checkPhraseForPassives: Checks phrases to determine whether the sentence is a passive sentence and adds this as a sentence-level feature. Uses sentence::checkPhraseForPassives recursively.

convertContractionsAndCommonAbbreviations: Converts a number of commonly appearing contractions and abbreviations into its fully written form.

compress: Compresses the sentence object, removing adjectives, adverbs, and function words.

createAppositiveSentences: Creates a sentence object for appositive clauses. Uses sentence::addToBeInFront and sentence::addSubjectToFront.

filterVariants: Throws away non-occurring combinations of variants based on the *n*-gram language model. Uses DBI::db::lookup.

findAgens: Searches for the agent in passive sentences.

findAllVerbs: Finds all verbs in the sentence object and clusters them into a verb group. Checks for verbs in phrase objects using sentence::findAllVerbsInPhrase.

findAllVerbsInPhrase: Finds all verbs in the phrase object and clusters them into a verb group. Checks for verbs in phrases recursively using sentence::findAllVerbsInPhrase.

findNonWords: Determines which tokens are non-words and which ones are real words. Uses word::lookupInDictionary.

findSeparableVerbs: Activated when Alpino is disabled or fails. Detects separable verbs in the calling sentences. Uses information from Vandeghinste (2002) to decide whether the most plausible candidate for a separable verb is more likely than keeping the parts apart.

findVariants: Generates variants for non-words or infrequent real words. Pushes these variants into the {spellingalternatives} feature. Uses word::findPhoneticVariants, word::WordSplitter, word::findOneInsertion, word::findOneDeletion, word::findOneSubstitution, and word::findOneTransposition.

fuzzymatch: Uses fuzzy matching techniques to select the best spelling hypothesis. Uses sentence::retrieveLines.

generateAlternatives: Generates different alternatives for the calling word object. Returns a reference to an array of arrays of word objects. Uses word::generateAlternatives.

generateEndOfSentence: Returns a word object containing </s> as lemma and as token.

generateProperText: Turns the message from Beta/Sclera into proper natural language text using the language model. Uses sentence::generateAlternatives, sentence::generateStartOfSentence, sentence::generateEndOfSentence, object::openNGramDatabase, pictoPath::containsAllInfo, sentence::sortSentencePathQ, sentencePath::hypothesisSolved, and sentencePath::printOutput.

generateStartOfSentence: Returns a word object containing <s> as lemma and as token.

generateTimePicto: Generates the appropriate temporal picto.

getWords: Recursively looks for word objects in phrases. Uses sentence::getWords.

HTMLOut: Provides HTML output of the message, pointing to picto images. Uses beta/sclera::getURL and beta/sclera::getPictoDirs.

JSONOut: Provides JSON output of the message.

lemmatize: Lemmatises each of the words in the sentence. Uses word::lemmatize.

makeActive: Converts a passive sentence into an active one by swapping the patient and the agent and removing the auxiliary. Uses sentence::findAgens.

makeSplitSentences: Splits the hypotheses sentences into characters and pushes them into the {splitsentences} feature. This is the pre-processing step for the character-based fuzzy matching.

moveAllVerbsBehindSubject: Moves the verb group behind the subject, if any.

moveAllVerbsToFront: Moves the verb group to the front of the sentence.

moveInterrogativeVerbs: Moves verb groups in interrogative clauses. Uses sentence::moveAllVerbsBehindSubject and sentence::addQuestionWordToFront.

moveMainAndSSUBVerbs: Moves verb groups in main, OTI, and SSUB clauses. Uses sentence::addSubjectToFront and sentence::moveAllVerbsBehindSubject.

movePPsToBack: Moves PPs and adverbs at the beginning of the sentence (in front of the subject) to the back of the sentence.

moveRelObjectVerbs: Moves verb groups in relative clauses that have an (in)direct object function.

Uses sentence::moveAllVerbsBehindSubject and sentence::addObjectBehindVerbs.

moveRelSubjectVerbs: Moves verb groups in relative clauses that have a subject function or PPRES clauses. Uses sentence::moveAllVerbsToFront and sentence::addSubjectToFront.

ParallelJSONOut: Provides parallel text and JSON output of the message.

ParallelOutOtherFile: Creates a temporary file for generating parallel JSON output.

removeCapitalFirstWord: Lowercases the first word of each sentence.

removeDoublePaths: Removes doubles in the list of paths. Uses pictopath::stringify and pictopath::weight.

removeFlooding: Tackles flooding. When more than 2 identical characters appear in sequence, they are reduced to a maximum of 2 characters.

retrieveLines: Retrieves the lines from the fuzzy matching output file and calculates the winning hypothesis. Determines whether a highly similar character match is found. If this is the case, performs new substitutions in the hypothesis.

searchArgMax: Looks for the optimal path to convert the message into pictos. Uses sentence::removeDoublePaths and sentence::SortQ.

sortQ: Sorts an array of picto paths by their weight. Uses pictopath::weight.

sortSentencePathQ: Sorts an array of sentence paths by their cost.

TextOut: Provides textual output of the sentence.

G.2.4 The Phrase Class

ISA("object")

No methods apply to the phrase class.

G.2.5 The Word Class

ISA("object")

adaptToPolarity: For every word that has a negative polarity (as marked under the {polarity} feature), it is looked up whether, for the pictos attached to this word, there exists a negative picto in the database.

addLexUnits: Adds lexical units retrieved from the WordNet database. Looks up the lemma in the database and creates a new lexunit object for each matching lexical unit. Returns "undefined" when no lexical units are found.

addPictos: Adds picto objects to a word. Uses lexunit::addPicto for each lexical unit in the word. If no lexical units are present, uses word::addPictosNotInWordnet.

addPictosNotInWordnet: Adds pictos which are not represented in the lexical semantic database.

addSynsets: Adds the synsets to the object. Uses lexunit::addSynset for each lexunit object. If lexunit objects have not yet been added, word::addLexUnits is called first.

addWordnet: Checks if the word matches a noun, verb, adjective, or adverb tag. Uses word::addLexUnits and word::addSynsets.

checkDictionary: Checks if a picto is found in the dictionary. If so, retrieves an array of lemmas that are found in the dictionary.

checkPronouns: Checks if a picto is found in the list of pronouns. If so, retrieves an array of associated pronouns.

containsNoun: Checks whether the alternative (inflected form) is a noun. Uses word::isNoun.

endOfSentence: Activated when Alpino is disabled or fails. Checks if the calling word object is an end of sentence symbol. Returns 1 or "undefined".

filterLexUnitsAccordingToPos: Removes lexunits from list when part-of-speech tag does not match WordNet part-of-speech tag.

filterPos: Checks whether the part-of-speech tags match the part-of-speech tag of the word object. Returns the part-of-speech tags that have been matched.

findMostFrequentDepLemma: Finds the most frequent lemma in an array of lemmas of dependent synsets, according to a lexicon.

findMostFrequentLemma: Finds the most frequent lemma in an array of lemmas of single or head synsets, according to a lexicon.

findOneDeletion: Checks for a non-word or an infrequent real word whether a character deletion causes a real word with a minimum frequency to be created.

findOneInsertion: Checks for a non-word or an infrequent real word whether a character insertion causes a real word with a minimum frequency to be created.

findOneSubstitution: Checks for a non-word or an infrequent real word whether a character substitution causes a real word with a minimum frequency to be created.

findOneTransposition: Checks for a non-word or an infrequent real word whether an adjacent character transposition causes a real word with a minimum frequency to be created.

findPhoneticVariants: Checks for every non-word and low-frequency real word whether a real-word phonetic variant can be generated by splitting it into individual character sequences and replacing them by phonetically similar character sequences.

generateAlternatives: Generates different alternatives for the calling word object. Returns a reference to an array of arrays of word objects. Uses word::getAlternativeWordObjects, picto::getAlternativeWordObjects, word::lookupFileName, word::getString, word::containsNoun, and word::getArticles.

getAlternativeWordObjects: Generates different alternatives for the calling word object. Returns the paradigm of the word. Uses word::getParadigm.

getArticles: Returns matching articles for the calling word object.

getNegativeWord: Activated when Alpino is disabled or fails. Generates a word object for a negation.

getNumber: Returns 1 if the calling word object is singular and 2 if it is plural.

getParadigm: Activates a reverse lemmatiser, which generates a paradigm for the calling word object. Returns an array of possible alternatives.

getPictoAsDependents: Returns the pictos under the {picto_asdependent} feature of the calling word object.

getPictoComplexes: Returns the pictos under the {picto_complex} feature of the calling word object.

getPictoSingle: Returns the pictos under the {picto_single} feature of the calling word object.

getString: Converts alternatives into a string.

getSynsets: Gets the synset for each of the lexunits of the object. Returns an array.

isContentWord: Checks whether the word is a content word. If the file name of the word exists, the word is by definition a content word. Returns 1 or "undefined". Uses word::lookupFilename.

isNegative: Activated when Alpino is disabled or fails. Checks whether the calling word object is a negative. Returns 1 or "undefined".

isNeuter: Checks whether the calling word object is a neuter. Returns 1 or "undefined".

isNonNeuter: Checks whether the calling word object is a non-neuter. Returns 1 or "undefined".

isNoun: Checks whether the calling word object is a noun. Returns 1 or "undefined".

isPlural: Checks whether the calling word object is a plural. Returns 1 or "undefined".

lemmatize: Activated when Alpino is disabled or fails. Lemmatises a tagged word. Uses a list to lemmatise the word, based on the token and the part-of-speech tag. Uses word::lemmatize_rules.

lemmatize_rules: Activated when Alpino is disabled or fails. Lemmatises a tagged word. Uses a set of rules to lemmatise the word, based on the token and the part-of-speech tag.

lemmatizeAssociations: Lemmatises the associations that were retrieved by the semantic association tool for single or head synsets.

lemmatizeDepAssociations: Lemmatises the associations that were retrieved by the semantic association tool for dependent synsets.

lookupFilename: Looks up whether picto token + picture extension exists. If this is the case, adds a picto object with the file name of the lemma to the {picto_single} feature of the calling word. Returns 1 or "undefined". Uses beta/sclera::getPictoDirs and object::pushFeature.

lookupInDictionary: Checks if a token occurs in the lexicon. Adds a value to the {spellcheck} feature.

lookupPictoDictionary: Prepares the queries for word::lookupPictoDictionaryTokLemTag. Returns 1 or "undefined" if no picto is found.

lookupPictoDictionaryTokLemTag: Looks up the word in the picto dictionary and adds the picto object to the {picto_single} attribute. Returns 1 or "undefined" if no picto is found. Uses DBI::db::lookup.

makeDependentList: Creates a list of lemmas associated with a given dependent synset.

makeLemmaList: Creates a list of lemmas associated with a given single or head synset.

retrieveLemmas: Retrieves all lemmas from a picto. Uses word::makeLemmaList, word::makeDependentList, word::checkDictionary, and word::checkPronouns.

WordSplitter: Splits a non-word if its separate parts each form a real word with a minimum frequency.

G.2.6 The Lexunit Class

ISA("object")

addLemma: Given the ID in the lexunit object, retrieves the lemma from the WordNet database, and returns it.

addPicto: Adds the pictos to the synsets of the calling lexunit object under the three features {picto_single}, {picto_complex}, and {picto_asdependent}. If the calling lexunit object does not have a synset, returns "undefined".

addSynset: Adds the synset to a lexunit object. Looks up the synset in the database and creates a new synset object. Uses synset::addPos.

getAlternativeWordObjects: Generates different alternatives for the words in the calling lexunit object. Returns the paradigm of the word. Uses word::getAlternativeWordObjects .

getLemma: Given the ID in the lexunit object, retrieves the lemma from the WordNet database, and returns it. Uses lexunit::addLemma.

getPictoComplexes: Returns the pictos under the {picto_complex} feature of the calling lexunit object.

getPictoSingle: Returns the pictos under the {picto_single} feature of the calling lexunit object.

G.2.7 The Synset Class

ISA("object")

addAntonyms: Looks up the near antonyms of the synset and adds them to the {antonyms} feature. Returns "undefined" if no antonyms are found. Uses DBI::db::lookup.

addHyperonyms: Looks up the hyperonyms of the synset and adds them to the {hyperonyms} feature. Returns "undefined" if no hyperonyms are found. Uses DBI::db::lookup.

addHyponyms: Looks up the hyponyms of the synset and adds them to the {hyponyms} feature. Returns "undefined" in no hyponyms are found. Uses DBI::db::lookup.

addLemmas: Adds the lemma to each of the lexunit objects of the synset. If lexunit objects are not yet added, synset::addLexUnits is called first, and lexunit::addLemma is called on each of these lexunit objects. Returns an array of lemmas.

addLexUnits: Looks up the lexunits belonging to the synset. Uses DBI::db::lookup.

addPicto: Adds the picto items that match the synset (single, complex, and as dependent). Uses synset::addPictoTypes and synset::addPictoToRelations.

addPictoAsDependent: Adds the picto for which the calling synset is a dependent. Uses DBI::db::lookup and beta/sclera::getExtension.

addPictoComplex: Adds the complex pictos to the calling synset. Uses DBI::db::lookup and beta/sclera::getExtension.

addPictoSingle: Adds the single pictos to the calling synset. Uses DBI::db::lookup, beta/sclera::getExtension, beta/sclera::getPictoDirs, and beta/sclera::negativePicto.

addPictoToRelations: Adds the relations and the pictos that are attached to these relations. Uses synset::addRelations and synset::addPicto.

addPictoTypes: Takes as input a reference to an array of types (single, complex, and as dependent) and a penalty. Adds the picto items according to the type that matches the synset. Uses synset::addPictoSingle, synset::addPictoComplex, and synset::addPictoAsDependent.

addPos: Looks up "posspecific" in WordNet database. Uses DBI::db::lookup.

addRelations: Adds the relations. Uses synset::addXPosNearSynonyms and synset::addHyperonyms.

addXPosNearSynonyms: Looks up the XPosNearSynonyms of the synset and adds them to the {xposnearsynonyms} feature. Returns "undefined" if none are found. Uses DBI::db::lookup.

equal: Checks whether synset and calling synset have the same ID. Returns 1 or "undefined".

getAlternativeWordObjects: Generates different alternatives for the words in the calling synset object. Returns the paradigm of the word. Uses word::getAlternativeWordObjects.

getLemmas: Given the ID in the lexunit object, retrieves the lemma from the WordNet database, and returns it. Uses lexunit::getLemma.

occursInLexunit: Looks up whether the calling synset occurs in the lexunit object. Uses synset::occursInSynset. Returns 1 or "undefined".

occursInSynset: If the synset and the calling synset are equal, returns 1. Else calls recursively for the hyperonyms of the calling synset. Uses synset::occursInSynset.

occursInWord: Looks up whether the synset occurs in the word object. Returns 1 or "undefined".

occursInWords: Looks up whether the synset occurs in the array of words. Returns 1 or "undefined".

G.2.8 The Picto Class

ISA("object")

adaptToPolarity: In case of negative polarity, adds a negation picto to the pictopath object. Uses beta/sclera::negativePicto.

addSynsets: Adds the synsets to the object. Uses DBI::db::lookup, synset::addLemmas, synset::addPos, and object::pushFeature.

allDepsOccur: Takes a reference to an array of words as input (containing the words that still need to be processed). Checks whether all the dependents from the picto occur in the array. Returns 1 or "undefined". Uses synset::occursInWords.

checkInDictionary: Checks whether the picto occurs in the dictionary. Uses beta/sclera::getDictionaryTableName, DBI::db::lookup, and word::getAlternativeWordObjects.

checkInPronouns: Checks whether the picto occurs in the list of pronouns.

existNegative: Checks whether there exists an antonym of the calling picto in the database. Adapts the picto if this is the case. Returns 1 or "undefined". Uses DBI::db::lookup and beta/sclera::getExtension.

getAlternativeWordObjects: Generates different alternatives for the words in the calling picto object. Returns the paradigm of the word. Uses synset::getAlternativeWordObjects and word::getAlternativeWordObjects.

headOccurs: Takes a reference to an array of words as input (words to process). Checks whether the head of the calling picto object occurs in any of the words. Returns 1 or "undefined". Uses synset::occursInWords.

replaceSynset: Finds and replaces synsets in the calling picto object.

G.2.9 The Beta and Sclera Classes

ISA("picto")

getDictionaryTableName: Gets to the table in the WordNet database which contains dictionary info for the picto language.

getExtension: Sets the file name extension for the pictos.

getPictoDirs: Gets the path to the directory containing the pictos.

getPronouns: Returns the appropriate picto language pronoun picto.

getURL: Sets the URL for the pictos (used in HTML output mode).

negativePicto: Contains the picto file name for negation.

G.2.10 The Pictopath Class

ISA("object")

clone: Clones the path and the array, but not the objects inside the arrays.

containsAllInfo: Checks whether {wordstoprocess} feature list is empty. Returns 1 or "undefined".

extend: Expands the current path by looking for pictos in the next {wordtoprocess}. Returns array of new paths. Uses word::getPictoComplexes, picto::allDepsOccur, pictopath::extendWithPicto, pictopath::spliceWordWithDep, object::spliceFromArray, word::getPictoAsDependents, picto::headOccurs, pictopath::spliceWordWithHead, and pictopath:extendNoPicto.

extendNoPicto: Extends the path when no picto is found, creating a text object containing the input token. Checks if the input word has a polarity value. If this is the case, the negative picto is pushed in the {display} feature. Returns the new path. Uses pictopath::clone and object::pushFeature.

extendWithPicto: Returns a new path, which is a clone of the calling path, extending the {display} feature with the display of the picto, and extending the {words} feature with the word from the input arguments. Checks whether to adapt the polarity of picto. Uses picto::adaptToPolarity, pictopath::clone, and object::pushFeature.

spliceWordWithDep: Splices the word containing the synset of the picto dependents from the {wordstoprocess} feature of the calling pictopath object. Uses synset::occursInWords and object::pushFeature.

spliceWordWithHead: Splices the word containing the synset of the picto head from the {wordsto-process} feature of the calling pictopath object. Uses synset::occursInWords and object::pushFeature.

stringify: Takes all the displays and returns the contents. Uses image::getContent.

weight: Calculates and returns the weight. Weight = number of words to process + sum of path lengths of the pictos in the {display} feature.

G.2.11 The Sentencepath Class

ISA("object")

containsAllInfo: Checks whether {toprocess} feature list is empty. Returns 1 or "undefined".

cost: Returns the cost of the sentence path.

extend: Expands the current path by looking for words in the next {toprocess}. Returns an array of new paths. Uses ngram::estimateProb.

hypothesisSolved: Pushes the hypothesis into the {processed} feature of the sentencepath object. Uses object::pushFeature.

printOutput: Prints the winning hypothesis.

G.2.12 The Ngram Class

ISA("object")

estimateProb: Estimates the probability using Katz backoff. Uses unigram::estimateProb, bigram::estimateProb, trigram::estimateProb, fourgram::estimateProb, and fivegram::estimateProb.

G.2.13 The Unigram to Fivegram Classes

ISA("ngram")

estimateProb: Estimates the probability using Katz backoff. Uses DBI::db::lookup.

getAlfa: Estimates the backoff probability mass (alfa). Uses DBI::db::lookup.

G.2.14 The DBI::db Class

ISA("object")

close: Closes the connection to the PostgreSQL database.

execute: Checks whether an SQL statement can be executed. Returns 1 or "undefined".

lookup: Looks up an SQL statement in the PostgreSQL database. Uses DBI::db::execute.

new: Takes database name, host, port, user, and password as input. Connects to the PostgreSQL database.

G.2.15 The Image Class

ISA("object")

getContent: Displays the picto files.

G.2.16 The Text Class

ISA("object")

getContent: Displays the labels of the picto files.

G.3 GitHub

All scripts can be downloaded on GitHub:
https://github.com/VincentCCL/Picto/releases/tag/3.0.1

This repository is maintained by Vincent Vandeghinste and Leen Sevens.

Bibliography

Agrawal, Rakesh, Sreenivas Gollapudi, Anitha Kannan & Krishnaram Kenthapadi. 2011. Enriching Textbooks with Images. In Bettina Berendt, Arjen de Vries, Wenfei Fan, Craig Macdonald, Iadh Ounis & Ian Ruthven (eds.), *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM)*, 1847–1856. Glasgow, UK.

Angrosh, Mandya, Tadashi Nomoto & Advaith Siddharthan. 2014. Lexico-syntactic text simplification and compression with typed dependencies. In Junichi Tsujii & Jan Hajic (eds.), *Proceedings of the 25th International Conference on Computational Linguistics: Technical Papers (COLING)*, 1996–2006. Dublin, Ireland: Association for Computational Linguistics.

Augustinus, Liesbeth. 2015. *Complement Raising and Cluster Formation in Dutch. A Treebank-supported Investigation*. Utrecht, The Netherlands: LOT Dissertation Series.

Aw, Ai Ti, Min Zhang, Juan Xiao & Jian Su. 2006. A Phrase-based Statistical Model for SMS Text Normalization. In Nicoletta Calzolari, Claire Cardie & Pierre Isabelle (eds.), *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, 33–40. Sydney, Australia: Association for Computational Linguistics.

Baker, Bruce. 1982a. Minspeak: A semantic compaction system that makes self-expression easier for communicatively disabled individuals. *Byte* 7(9). 186–202.

Baker, Bruce. 1982b. Minspeak in the 21st Century. URL: http://www.minspeak.com/ (accessed September, 2018).

Baker, Collin F., Charles J. Fillmore & John B. Lowe. 1998. The Berkeley FrameNet Project. In Christian Boitet & Pete Whitelock (eds.), *Proceedings of the 17th International Conference on Computational Linguistics - Volume 1 (COLING)*, 86–90. Quebec, Canada: Association for Computational Linguistics.

Bassil, Youssef & Mohammad Alwani. 2012. Context-sensitive Spelling Correction Using Google Web 1T 5-Gram Information. *Computer and Information Science* 5(3). 37–48.

Bautista, Susana, Raquel Hervás, Agustín Hernández-Gil, Carlos Martínez-Díaz, Sergio Pascua & Pablo Gervás. 2017. AraTraductor: Text to Pictogram Translation using Natural Language Processing Techniques. In Juan Manuel González-Calleros (ed.), *Proceedings of 18th International Conference of the Spanish Human Computer Interaction Association (Interacción)*, 8 pp. Cancun, Mexico.

Beail, Nigel & Katie Williams. 2014. Using Qualitative Methods in Research with People Who Have Intellectual Disabilities. *Journal of Applied Research in Intellectual Disabilities* 27(2). 85–96.

Beardon, Colin, Claire Dormann, Stuart Mealing & Masoud Yazdani. 1992. Talking with pictures: Exploring the possibilities of iconic communication. *Association for Learning Technology Journal* 1(1). 26–39.

Beaufort, Richard, Sophie Roekhaut, Louise-Amélie Cougnon & Cédrick Fairon. 2010. A hybrid rule/model-based finite-state framework for normalizing SMS messages. In Jan Hajič (ed.), *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, 770–779. Uppsala, Sweden: Association for Computational Linguistics.

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent & Christian Jauvin. 2003. A Neural Probabilistic Language Model. *Journal of Machine Learning Research* 3. 1137–1155.

Bhamidipaty, Anuradha & Deepak Padmanabhan. 2007. SymAB: Symbol-Based Address Book for the Semi-literate Mobile User. In Cecília Baranauskas, Philippe Palanque, Julio Abascal & Simone Diniz Junqueira Barbosa (eds.), *Proceedings of the 11th IFIP TC13 International Conference on Human-Computer Interaction (INTERACT)*, 389–392. Rio de Janeiro, Brazil.

Bhattacharya, Samit & Anupam Basu. 2009. Design of an Iconic Communication Aid for Individuals in India With Speech and Motion Impairments. *Assistive Technology* 21(4). 173–187.

Bigman, Alex. 2014. Gestalt Principles and the Psychology of Design. URL: https://www.sitepoint.com/gestalt-principles-psychology-design/ (accessed September, 2018).

Blattner, Meera M., Denise A. Sumikawa & Robert M. Greenberg. 1989. Earcons and Icons: Their Structure and Common Design Principles. *Human-Computer Interaction* 4(1). 11–44.

Bliss, Charles K. 1965. *Semantography*. Australia: Semantography Publications.

BNC. 2007. The British National Corpus, Version 3.

Bott, Stefan, Horacio Saggion & Simon Mille. 2012. Text Simplification Tools for Spanish. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*, 1665–1671. Istanbul, Turkey: European Language Resources Association.

Bulté, Bram, Leen Sevens & Vincent Vandeghinste. In press. Automating lexical simplification in Dutch. *Computational Linguistics in the Netherlands Journal* 8. 23 pp.

Busche, Laura. 2015. Simplicity, Symmetry and More: Gestalt Theory And The Design Principles It Gave Birth To. URL: https://www.canva.com/learn/gestalt-theory/ (accessed September, 2018).

Byrne, Michael D. 1993. Using Icons to Find Documents: Simplicity is Critical. In Bert Arnold, Gerrit van der Veer & Ted White (eds.), *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing System (INTERCHI)*, 446–453. Amsterdam, The Netherlands: ACM Press.

Böcker, Martin. 1996. A multiple index approach for the evaluation of pictograms and icons. *Computer Standards and Interfaces* 18(1). 107–115.

Callison-Burch, Chris, Miles Osborne & Philipp Koehn. 2006. Re-evaluating the Role of BLEU in Machine Translation Research. In Frank Keller & Gabor Proszeky (eds.), *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 249–256. Trento, Italy.

Cambridge, Paul & Michelle McCarthy. 2001. User focus groups and Best Value in services for people with learning disabilities. *Health and Social Care in the Community* 9(6). 476–489.

Candido Jr., Arnaldo, Erick Maziero, Caroline Gasperin, Thiago A.S. Pardo, Lucia Specia & Sandra M. Aluisio. 2009. Supporting the Adaptation of Texts for Poor Literacy Readers: a Text Simplification Editor for Brazilian Portuguese. In Joel Tetreault, Jill Burstein & Claudia Leacock (eds.), *Proceedings of the NAACL-HLT Workshop on Innovative Use of NLP for Building Educational Applications*, 34–42. Boulder, CO: Association for Computational Linguistics.

Carl, Michael & Andy Way. 2003. *Recent Advances in Example-based Translation*. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Carpuat, Marine & Dekai Wu. 2005. Word Sense Disambiguation vs. Statistical Machine Translation. In Kevin Knight, Hwee Tou Ng & Kemal Oflazer (eds.), *Proceedings of the 43th Annual*

Meeting on Association for Computational Linguistics (ACL), 387–394. Ann Arbor, MI: Association for Computational Linguistics.

Carroll, John, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin & John Tait. 1999. Simplifying Text for Language-Impaired Readers. In Harry Thompson & Alex Lascarides (eds.), *Proceedings of the 9th Conference of the European Chapter of the ACL (EACL)*, 269–270. Bergen, Norway: Association for Computational Linguistics.

Chandrasekar, Raman, Christine Doran & Bangalore Srinivas. 1996. Motivations and Methods for Text Simplification. In Junichi Tsujii (ed.), *Proceedings of the 16th Conference on Computational Linguistics - Volume 2 (COLING)*, 1041–1044. Copenhagen, Denmark: Association for Computational Linguistics.

Chandrasekar, Raman & Bangalore Srinivas. 1997. Automatic induction of rules for text simplification. *Knowledge-Based Systems* 10. 183–190.

Chen, Alex. 2004. *The Visual Messenger Project*: University of Washington dissertation.

Chen, Stanley F. & Joshua Goodman. 1999. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language* 17. 359–394.

Cho, Heeryon, Toru Ishida, Toshiyuki Takasaki & Satoshi Oyama. 2008. Assisting Pictogram Selection with Semantic Interpretation. In Sean Bechhofer, Manfred Hauswirth, Jörg Hoffmann & Manolis Koubarakis (eds.), *Proceedings of the 5th European Semantic Web Conference (ESWC)*, 65–79. Tenerife, Spain: Springer Berlin Heidelberg.

Cho, Heeryon, Toru Ishida, Naomi Yamashita, Rieko Inaba, Yumiko Mori & Tomoko Koda. 2007. Culturally-Situated Pictogram Retrieval. In Toru Ishida, Susan Fussell & Piek Vossen (eds.), *Proceedings of the First International Workshop on Intercultural Collaboration (IWIC)*, 221–235. Kyoto, Japan: Springer Berlin Heidelberg.

Cho, Heeryon, Toru Ishida, Naomi Yamashita, Tomoko Koda & Toshiyuki Takasaki. 2009. Human Detection of Cultural Differences in Pictogram Interpretations. In Susan Fussell, Pamela Hinds & Toru Ishida (eds.), *Proceedings of 2009 ACM International Workshop on Intercultural Collaboration (IWIC)*, 165–174. Palo Alto, CA: ACM Press.

Choudhury, Monojit, Rahul Saraf, Vijit Jain, Animesh Mukherjee, Sudeshna Sarkar & Anupam Basu. 2007. Investigation and Modeling of the Structure of Texting Language. *International Journal of Document Analysis and Retrieval: Special Issue on Analytics of Noisy Text* 10. 157–174.

Cowgill, Jamie & Jim Bolek. 2003. Symbol Usage in Health Care Settings for People with Limited English Proficiency: Part One - Evaluation of Use of Symbol Graphics in Medical Settings.

Coyne, Bob & Richard Sproat. 2001. WordsEye: An Automatic Text-to-Scene Conversion System. In Lynn Pocock (ed.), *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 487–496. New York, NY: Association for Computational Linguistics.

Daelemans, Walter, Anja Höthker & Erik Tjong Kim Sang. 2004. Automatic Sentence Simplification for Subtitling in Dutch and English. In Maria Teresa Lino, Maria Francisca Xavier, Fátima Ferreira, Rute Costa & Raquel Silva (eds.), *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*, 1045–1048. Lisbon, Portugal: European Language Resources Association.

Daems, Jo, Jan Dekelver, Annelies De Vliegher, Jolien Smets & Lien Martens. 2015. Able To Include: Focus groups with persons with IDD and their coaches communicating through social media. In Jan Engelen (ed.), *Proceedings of the Conference on Engineering4Society*, 76–79. Heverlee, Belgium: IEEE Computer Society Press.

Davies, Daniel K., Steven E. Stock, Larry R. King, R. Brian Brown, Michael L. Wehmeyer & Karrie A. Shogren. 2015. An Interface to Support Independent Use of Facebook by People With Intellectual Disability. *Intellectual and Developmental Disabilities* 53(1). 30–41.

De Clercq, Orphée, Sarah Schulz, Bart Desmet, Els Lefever & Véronique Hoste. 2013. Normalization of Dutch User-Generated Content. In Adam Przepiórkowski & Maciej Ogrodniczuk (eds.), *Proceedings of the 9th International Conference on Recent Advances in Natural Language Processing (RANLP)*, 179–188. Hissar, Bulgaria.

De Neef, Emilie Guimier & Sébastien Fessard. 2007. Evaluation d'un système de transcription de SMS. In Catherine Camugli, Matthieu Constant & Anne Dister (eds.), *26th Conference on Lexis and Grammar*, 217–224. Bonifacio, France.

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li & Li Fei-Feit. 2009. ImageNet: A large-scale hierarchical image database. In Patrick Flynn & Eric Mortensen (eds.), *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 248–255. Miami, FL: IEEE Computer Society Press.

Denkowski, Michael & Alon Lavie. 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Christoph Monz, Matt Post & Lucia Specia (eds.), *Proceedings* *of the EACL 2014 Workshop on Statistical Machine Translation (WMT)*, 376–380. Baltimore, MD: Association for Computational Linguistics.

Devlin, Siobhan & John Tait. 1998. The Use of a Psycholinguistic Database in the Simplification of Text for Aphasic Readers. In CSLI Publications (ed.), *Linguistic Databases*, 161–173. Stanford, CA: John Nerbonne.

Ding, Chaohai, Nawar Halabi, Lama Al-Zaben, Yunjia Li, E.A. Draffan & Mike Wald. 2015. A Web based Multi-Linguists Symbol-to-Text AAC Application. In Luis Carriço & Silvia Mirri (eds.), *Proceedings of the 12th Web for All Conference (W4A)*, 2 pp. Florence, Italy.

Doddington, George. 2002. Automatic Evaluation of Machine Translation Quality Using N-gram Co-occurrence Statistics. In Mitchell Marcus (ed.), *Proceedings of the Second International Conference on Human Language Technology Research (HLT)*, 138–145. San Diego, CA.

Draffan, E.A., Mike Wald, Nawar Halabi, Ouadie Sabia, Wajdi Zaghouani, Amatullah Kadous, Amal Idris, Nadine Zeinoun, David Banes & Dana Lawand. 2015. Generating acceptable Arabic Core Vocabularies and Symbols for AAC users. In Jan Alexandersson, Ercan Altinsoy, Heidi Christensen, Peter Ljunglöf, François Portet & Frank Rudzicz (eds.), *Proceedings of the 6th Workshop on Speech and Language Processing for Assistive Technologies (SLPAT)*, 91–96. Dresden, Germany: Interspeech.

Dreyfuss, Henry. 1972. *Symbol Sourcebook. An Authoritative Guide to International Graphic Symbols*. Columus, OH: McGraw Hill Higher Education.

Dusserre, Emmanuelle, & Muntsa Padró. 2017. Bigger does not mean better! We prefer specificity. In Claire Gardent & Christian Retoré (eds.), *Proceedings of the 12th International Conference on Computational Semantics (IWCS)*, 6 pp. Montpellier, France.

Fillmore, Charles J. 1982. Frame semantics. *Linguistics in the Morning Calm* 111–137.

Finch, Andrew Michael, Wei Song, Kumiko Tanaka-Ishii & Eiichiro Sumita. 2013. Speaking Louder than Words with Pictures Across Languages. *AI Magazine* 34. 31–47.

Finch, Andrew Michael, Wei Song, Kumiko Tanaki-Ishii & Eiichiro Sumita. 2011. picoTrans: Using Pictures as Input for Machine Translation on Mobile Devices. In Toby Walsh (ed.), *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2614–2619. Barcelona, Spain.

Fitrianie, Siska & Léon J.M. Rothkrantz. 2005. Communication in Crisis Situations using Icon Language. In Arnold Smeulders, Jan Biemand, Martin Kersten & Peter Werkhoven (eds.), *Proceedings of the 2005 IEEE International Conference on Multimedia and Expo (ICME)*, 1370–1373. Amsterdam, The Netherlands: IEEE Computer Society Press.

Fitrianie, Siska & Léon J.M. Rothkrantz. 2006. Two-Dimensional Visual Language Grammar. In Petr Sojka, Ivan Kopecek & Karel Pala (eds.), *Proceedings of the 9th International Conference on Text, Speech and Dialogue (TSD)*, 573–580. Brno, Czech Republic: Springer Berlin Heidelberg.

Freyhoff, Geert, Gerard Hess, Linda Kerr, Bror Tronbacke & Kathy Van Der Veken. 1998. Make it simple. Tech. rep. Cornell University ILR School Ithaca, NY.

García, Luís Felipe, Luís Caldas de Oliveira & David Martins de Matos. 2015. Evaluating pictogram prediction in a location-aware augmentative and alternative communication system. *Assistive Technology* 28(2). 83–92.

Gaver, William. 1986. Auditory Icons: Using Sound in Computer Interfaces. *Human–Computer Interaction* 2(2). 167–177.

Gittins, David. 1986. Icon-based human-computer interaction. *International Journal of Man-Machine Studies* 24(6). 519–543.

Goonetilleke, Ravindra S., Heloisa Martins Shih, Hung Kai On & Julien Fritsch. 2001. Effects of training and representational characteristics in icon design. *International Journal of Human-Computer Studies* 55(5). 741–760.

Graham, Yvette, Nitika Mathur & Timothy Baldwin. 2014. Randomized Significance Tests in Machine Translation. In Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Christoph Monz, Matt Post & Lucia Specia (eds.), *Proceedings of the Ninth Workshop on Statistical Machine Translation (WMT)*, 266–274. Baltimore, MD.

Green, Spence, Jeffrey Heer & Christopher D. Manning. 2013. The Efficacy of Human Post-Editing for Language Translation. In Wendy Mackay, Stephen Brewster & Susanne Bødker (eds.), *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 439–448. Paris, France.

Haeseryn, Walter, Kirsten Romijn, Guido Geerts, Jaap de Rooij & Maarten van den Toorn. 1997. *Algemene Nederlandse Spraakkunst - 2nd Edition*. Groningen, The Netherlands: Martinus Nijhoff.

Halácsy, Péter, András Kornai & Csaba Oravecz. 2007. HunPos - an open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics,* *Companion Volume, Proceedings of the Demo and Poster Sessions (ACL)*, 209–212. Prague, Czech Republic.

Han, Bo & Timothy Baldwin. 2011. Lexical Normalisation of Short Text Messages: Makn Sens a #Twitter. In Annie Zaenen & Antal van den Bosch (eds.), *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1 (ACL-HLT)*, 368–378. Portland, OR: Association for Computational Linguistics.

Hayes, Gillian R., Sen Hirano, Gabriela Marcu, Mohamad Monibi, David H. Nguyen & Michael Yeganyan. 2010. Interactive visual supports for children with autism. *Personal and Ubiquitous Computing* 14(7). 663–680.

Heafield, Kenneth, Ivan Pouzyrevsky, Jonathan Clark & Philipp Koehn. 2013. Scalable Modified Kneser-Ney Language Model Estimation. In Pascale Fung & Massimo Poesio (eds.), *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, 7 pp. Sofia, Bulgaria.

HOW Staff. 2015. Gestalt Theory in Typography and Design Principles. URL: http://www.howdesign.com/resources-education/online-design-courses-education/gestalt-theory-typography-design-principles/ (accessed September, 2018).

Huer, Mary B. 2000. Examining Perceptions of Graphic Symbols Across Cultures: Preliminary Study of the Impact of Culture/Ethnicity. *Augmentative and Alternative Communication* 16(3). 180–185.

Hunnicutt, Sheri. 1986. Bliss Symbol-to-Speech Conversion: Blisstalk. *Journal of the American Voice I/O Society* 3. 19–38.

Ide, Nancy & Keith Suderman. 2004. The American National Corpus First Release. In Maria Teresa Lino, Maria Francisca Xavier, Fátima Ferreira, Rute Costa & Raquel Silva (eds.), *Proceedings of the Fourth Conference on Language Resources and Evaluation (LREC)*, 1681–1684. Lisbon, Portugal.

Innocent, Troy. 2001. The Language of Iconica. *Leonardo* 34(3). 255–259.

Isherwood, Sarah. 2009. Graphics and Semantics: The Relationship between What Is Seen and What Is Meant in Icon Design. In Don Harris (ed.), *Proceedings of 8th International Conference on Engineering Psychology and Cognitive Ergonomics (EPCE)*, 197–205. San Diego, CA: Springer Berlin Heidelberg.

Isherwood, Sarah, Siné J.P. McDougall & Martin B. Curry. 2007. Icon Identification in Context: The Changing Role of Icon Characteristics With User Experience. *The Journal of the Human Factors and Ergonomics Society* 49(3). 465–476.

ISO, International Organization of Standards. 1984. ISO/TR 7239: Development and principles for application of public information symbols.

ISO, International Organization of Standards. 2010. ISO 9241-210 Ergonomics of human-system interaction, Part 210: Human-centred design for interactive systems.

Ivy, Sarah E., Deborah D. Hatton & Jonathan D. Hooper. 2014. Using the Picture Exchange Communication System With Students With Visual Impairment. *Exceptional Children* 80. 474–488.

Jacobs, Gilles. 2015. *Word sense disambiguation in text-to-pictograph conversion*: KU Leuven dissertation.

Jing, Hongyan. 1998. Usage of WordNet in Natural Language Generation. In Christian Boitet & Pete Whitelock (eds.), *Proceedings of the Joint 17th International Conference on Computational Linguistics 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, 128–134. Montréal, Canada.

Joshi, Aravind K., Leon S. Levy & Masako Takahashi. 1975. Tree Adjunct Grammars. *Journal of Computer and System Sciences* 10(1). 136–163.

Joshi, Aravind K. & Yves Schabes. 1997. Tree-adjoining grammars. In Grzegorz Rozenberg & Arto Salomaa (eds.), *Handbook of Formal Languages Vol. 3: Beyond Words*, 69–123. Berlin, Germany: Springer Berlin Heidelberg.

Joshi, Dhiraj, James Z. Wang & Jia Li. 2006. The Story Picturing Engine - A System for Automatic Text Illustration. *ACM Transactions on Multimedia Computing, Communications and Applications* 2(1). 68–89.

Just, Marcel Adam & Patricia A. Carpenter. 1992. A Capacity Theory of Comprehension: Individual Differences in Working Memory. *Psychological Review* 99(1). 122–149.

Keating, Thomas. 2006. Picture Planner: A Cognitively Accessible Personal Activity Scheduling Application. In Simeon Keates & Simon Harper (eds.), *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS)*, 239–240. Portland, Oregon.

Keskinen, Tuuli, Tomi Heimonen, Markku Turunen, Juha-Pekka Rajaniemi & Sami Kauppinen. 2012. SymbolChat: A flexible picture-based communication platform for users with intellectual disabilities. *Interacting with Computers* 24(1). 374–386.

Klein, Dan & Christopher Manning. 2003. Accurate Unlexicalized Parsing. In Erhard Hinrichs & Dan Roth (eds.), *Proceedings of the 41st Meeting of the Association for Computational Linguistics (ACL)*, 423–430. Sapporo, Japan: Association for Computational Linguistics.

Klein, Guillaume, Yoon Kim, Yuntian Deng, Jean Sennellart & Alexander Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *ArXiv E-prints* 6 pp.

Kobus, Catherine, François Yvon & Géraldine Damnati. 2008. Normalizing SMS: are two metaphors better than one? In Donia Scott & Hans Uszkoreit (eds.), *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1 (COLING)*, 441–448. Manchester, UK: Association for Computational Linguistics.

Koehn, Philipp. 2004. Statistical Significance Tests for Machine Translation Evaluation. In Dekang Lin & Dekai Wu (eds.), *Proceedings of 2004 Conference on Empirical Methods on Natural Language Processing (EMNLP)*, 8 pp. Barcelona, Spain: Association for Computational Linguistics.

Koehn, Philipp. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In Asia-Pacific Association for Machine Translation (ed.), *Proceedings of the 10th Machine Translation Summit (MT Summit)*, 79–86. Phuket, Thailand.

Koehn, Philipp. 2009. *Statistical Machine Translation*. Cambridge, MA: Cambridge University Press.

Koehn, Philipp. 2017. *Neural Machine Translation. URL: https://arxiv.org/pdf/1709.07809 (accessed September, 2018)*. Baltimore, MD: Johns Hopkins University.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin & Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *The Annual Meeting of the Association for Computational Linguistics, demonstration session (ACL)*, Prague, Czech Republic.

Kolb, Peter. 2008. DISCO: A Multilingual Database of Distributionally Similar Words. In Ernst Buchberger, Stefan Busemann, Alexander Geyken, Jost Gippert, Karin Harbusch, Rüdiger Hoffmann, Bernd Kröger, Lothar Lemnitzer, Sabine Schulte, Manfred Stede & Angelika Storrer (eds.), *Proceedings of the 9th Konferenz zur Verarbeitung Natürlicher Sprache (KONVENS)*, 8 pp. Berlin, Germany.

Lamy, Jean-Baptiste, Catherine Duclos, Avner Bar-Hen, Patrick Ouvrard & Alain Venot. 2008. An iconic language for the graphical representation of medical concepts. *BMC Medical Informatics and Decision Making* 8(16). 12 pp.

Landis, J. Richard & Gary G. Koch. 1977. The Measurement of Observer Agreement for Categorical Data. *Biometrics* 33(1). 159–174.

Leemans, Paul. 2001. *VIL: A Visual Inter Lingua*: Worcester Polytechnic Institute dissertation.

Lemmens, Adriaan. 2016. *Rule-based pictograph-to-text translation with Depicto 0.1*: KU Leuven dissertation.

Levenshtein, Vladimir. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady* 10(8). 707–710.

Li, Jianhua & Graeme Hirst. 2005. Semantic Knowledge in Word Completion. In Andrew Sears & Enrico Pontelli (eds.), *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS)*, 121–128. Baltimore, MD: ACM Press.

Lidwell, William, Kritina Holden & Jill Butler. 2003. *Universal Principles of Design*. Gloucester, MA: Rockport Publishers.

Lin, Chin-Yew. 2004. ROUGE: a Package for Automatic Evaluation of Summaries. In Marie-Francine Moens & Stan Szpakowicz (eds.), *Proceedings of the Workshop on Text Summarization Branches Out*, 8 pp. Barcelona, Spain.

Lison, Pierre & Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting Large Parallel Corpora from Movie and TV Subtitles. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*, 923–929. Portorož, Slovenia: European Language Resources Association.

Liu, Fu-Hua, Liang Gu & Michael Picheny. 2003. Use of Statistical N-gram Models in Natural Language Generation for Machine Translation. In The Institute of Electrical & Electronics Engineers (eds.), *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 636–639. Hong Kong, Hong Kong.

Lodding, Kenneth. 1983. Iconic Interfacing. *IEEE Computer Graphics and Applications* 3(2). 11–20.

Ma, Xiaojuan & Perry R. Cook. 2009. How Well do Visual Verbs Work in Daily Communication for Young and Old Adults? In Daniel Olsen & Richard Arthur (eds.), *Proceedings of the 27th International Conference on Human Factors in Computing System (CHI)*, 361–364. Boston, MA: ACM Press.

Ma, Xiaoyue, Nada Matta, Jean-Pierre Cahier, Chunxiu Qin & Yanjie Cheng. 2015. From action icon to knowledge icon: Objective-oriented icon taxonomy in computer science. *Displays* 39. 68–79.

Maiti, Santa, Debasis Samanta, Satya R. Das & Monalisa Sarma. 2011. Language Independent Icon-Based Interface for Accessing Internet. In Jyotsna Kumar Mandal, Suresh Chandra Satapathy, Manas Kumar Sanyal & Vikrant Bhateja (eds.), *Proceedings of the First International Conference on Advances in Computing and Communications (ACC)*, 172–182. Kochi, India: Springer Berlin Heidelberg.

Manber, Udi & Gene Myers. 1993. Suffix Arrays: A New Method for On-line String Searches. *SIAM Journal on Computing* 22(5). 935–948.

Marcus, Aaron. 2003. Icons, Symbols, and Signs: Visible Languages to Facilitate Communication. *Interactions* 10(3). 37–43.

Martínez-Santiago, Fernando, Manuel Carlos Díaz-Galiano, Luis Alfonso Ureña-López & Ruslan Mitkov. 2015. A semantic grammar for beginning communicators. *Knowledge-Based Systems* 86. 158–172.

Mason, Jana M. & Janet Ross Kendall. 1979. Facilitating Reading Comprehension through Text Structure Manipulation. *Alberta Journal of Educational Research* 25(2). 68–76.

McCallum, Andrew Kachites. 2002. MALLET: A Machine Learning for Language Toolkit. URL: http://mallet.cs.umass.edu (accessed September, 2018).

McCoy, Kathleen F. 1998. Interface and Language Issues in Intelligent Systems for People with Disabilities. In Vibhu O. Mittal, Holly A. Yanco, John Aronis & Richard Simpson (eds.), *Assistive Technology and Artificial Intelligence: Applications in Robotics, User Interfaces and Natural Language Processing*, 11 pp. Berlin, Germany: Springer Berlin Heidelberg.

McCoy, Kathleen F., Christopher A. Pennington & Arlene Luberoff Badman. 1998. Compansion: From Research Prototype to Practical Integration. *Natural Language Engineering* 4(1). 73–95.

McCrum-Gardner, Evie. 2008. Which is the correct statistical test to use? *British Journal of Oral and Maxillofacial Surgery* 46(1). 38–41.

Medina Maestro, Jaime, Horacio Saggion, Ineke Schuurman, Leen Sevens, John O'Flaherty, Annelies De Vliegher & Jo Daems. 2016. Towards Integrating People with Intellectual Disabilities in the Digital World. In Emile Aarts, Hamid Aghajan, Michael Berger, Vic Callaghan, Diane Cook, Sajal Das, Anind Dey, Sylvain Giroux, Pertti Huuskonen, Jadwiga Indulska, Achilles Kameas, Peter Mikulecký, Daniel Shapiro, Toshiyo Tamura & Michael Weber (eds.), *Proceedings of the 7th International Workshop on Intelligent Environments Supporting Healthcare and Well-being (WISHWell)*, 348–357. London, UK.

Mihalcea, Rada & Chee Wee Leong. 2008. Toward Communicating Simple Sentences Using Pictorial Representations. *Machine Translation* 22(3). 153–173.

Miller, George A. 1995. WordNet: A Lexical Database for English. *Communications of the ACM* 28(11). 39–41.

Miller, Hannah. 2016. Investigating the Potential for Miscommunication Using Emoji. URL: https://grouplens.org/blog/investigating-the-potential-for-miscommunication-using-emoji/ (accessed September, 2018).

Miller, Hannah, Jacob Thebault-Spieker, Shuo Chang, Isaac Johnson, Loren Terveen & Brent Hecht. 2016. "Blissfully happy" or "ready to fight": Varying Interpretations of Emoji. In Krishna Gummadi & Markus Strohmaier (eds.), *Proceedings of the 10th International Conference on Web and Social Media (ICWSM)*, 259–268. Cologne, Germany: AAAI Press.

Mullachery, Vikram & Vishal Motwani. 2018. Image Captioning. *Computing Research Repository* 7 pp.

Munemori, Jun, Taro Fukuda, Moonyati Binti Mohd Yatid, Tadashi Nishide & Junko Itou. 2010. Pictograph Chat Communicator III: A Chat System That Embodies Cross-Cultural Communication. In Rossitza Setchi, Ivan Jordanov, Robert Howlett & Lakhmi Jain (eds.), *Proceedings of the 14th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems (KES)*, 473–482. Cardiff, UK: Springer Berlin Heidelberg.

Munenori, Jun, Tadashi Nishide, Tomoki Fujita & Junko Itou. 2011. Development of a Distributed Pictograph Chat System: Pictograph Chat Communicator IV. In Andreas König, Andreas Dengel, Knut Hinkelmann, Koichi Kise, Robert J. Howlett & Lakhmi C. Jain (eds.), *Knowledge-Based and Intelligent Information and Engineering Systems*, 77–85. Berlin, Germany: Springer Berlin Heidelberg.

Müller, Ingrid, Margret Buchholz & Ulrika Ferm. 2010. Text messaging with picture symbols – experiences of seven persons with cognitive and communicative disabilities. *Journal of Assistive Technologies* 4(4). 11–23.

Nakamura, Carlos & Qing Zeng-Treitler. 2012. A taxonomy of representation strategies in iconic communication. *International Journal of Human-Computer Studies* 70(8). 535–551.

Navigli, Roberto. 2009. Word Sense Disambiguation: A Survey. *ACM Computing Surveys* 41(2). 69 pp.

Navigli, Roberto & Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence* 193. 217–250.

Neale, Steven, Luís Gomes, Eneko Agirre, Oier Lopez de Lacalle & António Branco. 2016. Word Sense-Aware Machine Translation: Including Senses as Contextual Features for Improved Translation Models. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*, 2777–2783. Portorož, Slovenia: European Language Resources Association.

Nikolow, Sybilla. 2011. 'Words Divide, Pictures Unite.' Otto Neurath's Pictorial Statistics in Historical Context. In Richard Heinrich, Elisabeth Nemeth, Wolfram Pichler & David Wagner (eds.), *Image and Imaging in Philosophy, Science and the Arts, volume 2*, 85–98. Frankfurt, Germany: Ontos Verlag.

Nind, Melanie. 2009. Conducting qualitative research with people with learning, communication and other disabilities: Methodological challenges. Tech. rep. University of Southampton UK.

Noens, Ilse & Ina Berckelaer-Onnes. 2002. Communicatie bij mensen met autisme en een verstandelijke beperking: Van inzicht naar interventie. *Nederlands Tijdschrift voor de Zorg aan Mensen met een Verstandelijke Handicap* 28. 212–225.

Nomura, Misako, Gyda Skat Nielsen & Bror Tronbacke. 2010. Guidelines for Easy-to-Read Materials. URL: https://www.ifla.org/files/assets/hq/publications/professional-report/120.pdf (accessed September, 2018).

Och, Franz Josef & Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics* 29(1). 19–51.

Olah, Christopher. 2015. Understanding LSTM Networks. URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (accessed September, 2018).

Olaszi, Péter, Ilona Koutny & Sophia L. Kálmán. 2002. From Bliss Symbols to Grammatically Correct Voice Output: A Communication Tool for People with Disabilities. *International Journal of Speech Technology* 5(1). 49–56.

Oostdijk, Nelleke, Wim Goedertier, Frank Van Eynde, Louis Boves, Jean-Pierre Martens, Michael Moortgat & Harald Baayen. 2002. Experiences from the Spoken Dutch Corpus Project. In Manuel González Rodríguez & Carmen Paz Suarez Araujo (eds.), *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC)*, 340–347. Las Palmas, Canary Islands: European Language Resources Association.

Oostdijk, Nelleke, Martin Reynaert, Véronique Hoste & Henk van den Heuvel. 2013. SoNaR User Documentation. URL: https://ticclops.uvt.nl/SoNaR_end-user_documentation_v.1.0.4.pdf (accessed September, 2018), 65 pp.

Orasan, Constantin, Richard Evans & Ruslan Mitkov. 2018. Intelligent Text Processing to Help Readers with Autism. *Intelligent Natural Language Processing: Trends and Applications* 713–740.

Packard, Woodley. 2015. ACE, the Answer Constraint Engine. URL: http://sweaglesw.org/linguistics/ace/ (accessed September, 2018).

Papineni, Kishore, Salim Roukos, Todd Ward & Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In Pierre Isabelle (ed.), *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, 311–318. Philadelphia, PA: Association for Computational Linguistics.

Patel, Rupal, Sam Pilato & Roy Deb. 2004. Beyond Linear Syntax: An Image-Orientation Communication Aid. *Assistive Technology Outcomes and Benefits* 1(1). 57–66.

Pedersen, Ted, Satanjeev Banerjee & Siddharth Patwardhan. 2005. Maximizing semantic relatedness to perform word sense disambiguation. Tech. rep. University of Minnesota Supercomputing Institute Minnesota, MN.

Peirce, Charles S. 1867. On A New List of Categories. URL: http://www.peirce.org/writings/p32.html (accessed September, 2018).

Peirce, Charles S. 1908. Letter to Lady Welby-Gregory. URL: http://www.unav.es/gep/Welby12.10.04.html (accessed September, 2018).

Pereira, Suzanne, Sylvain Hassler, Saliha Hamek, César Boog, Nicolas Leroy, Beuscart-Zéphirm Marie Catherine, Madeleine Favre, Alain Venot, Catherine Duclos & Jean-Baptiste

Lamy. 2014. Improving access to clinical practice guidelines with an interactive graphical interface using an iconic language. *BMC Medical Informatics and Decision Making* 14(77). 13 pp.

Peters, Carol & Martin Braschler. 2001. European Research letter: Cross-language System Evaluation: the CLEF Campaigns. *Journal of the American Society for Information Science and Technology* 52(12). 1067–1072.

Pighin, Daniele, Lluís Formiga & Lluís Màrquez. 2012. A Graph-based Strategy to Streamline Translation Quality Assessments. In George Foster, Ray Flournoy, Mike Dillinger, Chuck Simmons & Nick Bemish (eds.), *Proceedings of the Tenth Conference of the Association for Machine Translation in the Americas (AMTA)*, 10 pp. San Diego, CA.

Pollard, Carl & Ivan A. Sag. 1994. *Head-driven Phrase Structure Grammar*. Chicago, IL: University of Chicago Press.

Poots, Kent J. & Ebrahim Bagheri. 2018. Automatic Annotation of Text with Pictures. *IT Professional* 20(1). 36–44.

Purchase, Helen. 1998. Defining Multimedia. *IEEE Multimedia* 5(1). 8–15.

Rogers, Yvonne. 1989. Icons at the interface: their usefulness. *Interacting with Computers* 1(1). 105–117.

Romski, MaryAnn, Rose A. Sevcik, Michael Smith, Ashlyn R. Barker, Stephanie Folan & Andrea Barton-Hulsey. 2009. The System for Augmenting Language: Implications for Young Children with Autism Spectrum Disorders. In Pat Mirenda & Teresa Iacono (eds.), *Autism Spectrum Disorders and AAC*, 219–245. Baltimore, MD: Paul H. Brookes Publishing Company.

Rowda, Juan & Olga Pospelova. 2016. Human Evaluation of Machine Translation. URL: https://www.gala-global.org/blog/human-evaluation-machine-translation (accessed September, 2018).

Saggion, Horacio, Elena Gómez-Martínez, Esteban Etayo, Alberto Anula & Lorena Bourg. 2011. Text Simplification in Simplext. Making Text More Accessible. *Procesamiento del Lenguaje Natural* 47. 341–342.

Saussure, Ferninand De. 1916. *Cours de linguistique générale*. Payot, Lausanne.

Schmid, Helmut. 1995. Improvements in Part-of-speech Tagging with an Application to German. In David Yarowsky & Kenneth Church (eds.), *Proceedings of the EACL-95 SIGDAT Workshop*, Dublin, Ireland.

Schulz, Sarah, Guy De Pauw, Orphée De Clercq, Bart Desmet, Véronique Hoste, Walter Daelemans & Lieve Macken. 2016. Multimodular Text Normalization of Dutch User-Generated Content. *ACM Transactions on Intelligent Systems and Technology* 7(4). 22 pp.

Scott, Derek. 1993. Visual search in modern human-computer interfaces. *Behaviour and Information Technology* 12(3). 174–189.

Sevens, Leen, Jo Daems, Annelies De Vliegher, Ineke Schuurman, Vincent Vandeghinste & Frank Van Eynde. 2017a. Building an Accessible Pictograph Interface for Users with Intellectual Disabilities. In Peter Cudd & Luc de Witte (eds.), *Proceedings of the 2017 Association for the Advancement of Assistive Technology in Europe Congress (AAATE)*, 870–877. Sheffield, UK: IOS Press.

Sevens, Leen, Gilles Jacobs, Vincent Vandeghinste, Ineke Schuurman & Frank Van Eynde. 2016a. Improving Text-to-Pictograph Translation Through Word Sense Disambiguation. In Claire Gardent, Raffaella Bernardi & Ivan Titov (eds.), *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics (*SEM)*, 131–135. Berlin, Germany.

Sevens, Leen, Tom Vanallemeersch, Ineke Schuurman, Vincent Vandeghinste & Frank Van Eynde. 2016b. Automated Spelling Correction for Dutch Internet Users with Intellectual Disabilities. In Ineke Schuurman, Vincent Vandeghinste & Horacio Saggion (eds.), *Proceedings of 1st LREC Workshop on Improving Social Inclusion using NLP: Tools and Resources (ISI-NLP)*, 11–19. Portorož, Slovenia.

Sevens, Leen, Vincent Vandeghinste, Ineke Schuurman & Frank Van Eynde. 2015a. Extending a Dutch Text-to-Pictograph Converter to English and Spanish. In Jan Alexandersson, Ercan Altinsoy, Heidi Christensen, Peter Ljunglöf, François Portet & Frank Rudzicz (eds.), *Proceedings of 6th Workshop on Speech and Language Processing for Assistive Technologies (SLPAT)*, 110–117. Dresden, Germany.

Sevens, Leen, Vincent Vandeghinste, Ineke Schuurman & Frank Van Eynde. 2015b. Natural Language Generation from Pictographs. In Anya Belz, Albert Gatt, François Portet & Matthew Purver (eds.), *Proceedings of 15th European Workshop on Natural Language Generation (ENLG)*, 71–75. Brighton, UK.

Sevens, Leen, Vincent Vandeghinste, Ineke Schuurman & Frank Van Eynde. 2017b. Simplified Text-to-Pictograph Translation for People with Intellectual Disabilities. In A Ittoo, Le Minh Nguyen & Elisabeth Métais (eds.), *Natural Language Processing and Information Systems: 22nd International Conference on Applications of Natural Language to Information Systems*, 185–196. Springer International Publishing.

Sevens, Leen, Vincent Vandeghinste, Ineke Schuurman & Frank Van Eynde. In press. Less is more: A rule-based syntactic simplification module for improved text-to-pictograph translation. *Data & Knowledge Engineering* 26 pp.

Sevens, Leen, Vincent Vandeghinste & Frank Van Eynde. 2014. Improving the Precision of Synset Links Between Cornetto and Princeton WordNet. In Jorge Baptista, Pushpak Bhattacharyya, Christiane Fellbaum, Mikel Forcada, Chu-Ren Huang, Svetla Koeva, Cvetana Krstev & Eric Laporte (eds.), *Proceedings of the Workshop on Lexical and Grammatical Resources for Language Processing (LG-LP)*, 120–126. Dublin, Ireland.

Shiwen, Yu. 1993. Automatic evaluation of output quality for Machine Translation systems. *Machine Translation* 8. 117–126.

Siddharthan, Advaith. 2006. Syntactic Simplification and Text Cohesion. *Research On Language And Computation* 4(1). 77–109.

Siddharthan, Advaith. 2014. A survey of research on text simplification. *International Journal of Applied Linguistics* 165(2). 259–298.

Skadins, Raivis, Jörg Tiedemann, Roberts Rozis & Daiga Deksne. 2014. Billions of Parallel Words for Free: Building and Using the EU Bookshop Corpus. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk & Stelios Piperidi (eds.), *Proceedings of the Ninth International Confererence on Language Resources and Evaluation (LREC)*, 1050–1055. Reykjavik, Iceland: European Language Resources Association.

Skovholt, Karianne, Anette Grønning & Anne Kankaanranta. 2014. The Communicative Functions of Emoticons in Workplace E-Mails: :-). *Journal of Computer-Mediated Communication* 19. 780–797.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla & John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In Laurie Gerber (ed.), *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas (AMTA)*, 223–231. Cambridge, MA.

Soares, Marcos André Barroso. 2015. *Designing Culturally Sensitive Icons for User Interfaces: An approach for the Interaction Design of smartphones in developing countries*: Universidade do Porto dissertation.

Sobrevilla Cabezudo, Marco Antonio, Nora La Serna Palomino & Rolando Maguiña Perez. 2015. Improving subjectivity detection for Spanish texts using subjectivity word sense disambiguation based on knowledge. In Hector Cancela, Alex Cuadros-Vargas & Ernesto Cuadros-Vargas (eds.), *Proceedings of the 2015 Latin American Computing Conference (CLEI)*, 7 pp. Arequipa, Peru.

Somers, Harold. 2007. Theoretical and methodological issues regarding the use of language technologies for patients with limited English proficiency. In Andy Way & Barbara Gawronska (eds.), *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-07)*, 206–213. Skövde, Sweden.

Song, Wei, Andrew Finch, Kumiko Tanaka-Ishii, Keiji Yasuda & Eiichiro Sumita. 2013. picoTrans: An Intelligent Icon-driven Interface for Cross-lingual Communication. *ACM Transactions on Interactive Intelligent Systems* 3(1). 31 pp.

Speer, Robert & Catherine Havasi. 2012. Representing General Relational Knowledge in ConceptNet 5. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*, 3679–3686. Istanbul, Turkey: European Language Resources Association.

Sproat, Richard, Alan W. Black, Stanley Chen, Shankar Kumar, Mari Ostendorf & Christopher Richards. 2001. Normalization of non-standard words. *Computer Speech and Language* 15(3). 287–333.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever & Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15. 1929–1958.

Steinberger, Ralf, Andreas Eisele, Szymon Klocek, Spyridon Pilos & Patrick Schlüter. 2012. DGT-TM: A freely Available Translation Memory in 22 Languages. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*, 454–459. Istanbul, Turkey: European Language Resources Association.

Stergiou, Christos & Dimitrios Siganos. 2018. Neural Networks. URL: https://www.doc.ic.ac.uk/ nd/surprise_96/journal/vol4/cs11/report.html (accessed September, 2018).

Stocky, Tom, Alexander Faaborg & Henry Liebermann. 2004. A Commonsense Approach to Predictive Text Entry. In Elizabeth Dykstra-Erickson & Manfred Tscheligi (eds.), *Proceed-*

ings of the Conference on Human Factors in Computing Systems (CHI), 1163–1166. Vienna, Austria: ACM Press.

Stolcke, Andreas. 2002. SRILM: An Extensible Language Modeling Toolkit. In John Hansen & Bryan Pellom (eds.), *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP)*, 901–904. Denver, CO.

Stoop, Wessel & Antal van den Bosch. 2014. Improving word prediction for augmentative communication by using idiolects and sociolects. *Dutch Journal of Applied Linguistics* 3(2). 137–154.

Swiffin, Andrew L., J. Adrian Pickering, John L. Arnott & Alan F. Newell. 1985. PAL: An effort efficient portable communication aid and keyboard emulator. In Geoffrey Bart Thrope & Jorge Ernesto Letechipia (eds.), *Proceedings of the Eighth Annual Conference on Rehabilitation Technology*, 197–199. Memphis, TN: Rehabilitation Engineering Society of North America.

Takasaki, Toshiyuki. 2006. PictNet: Semantic Infrastructure for Pictogram Communication. In Petr Sojka, Key-Sun Choi, Christiane Fellbaum & Piek Vossen (eds.), *Proceedings of the The Third International WordNet Conference (GWC)*, 279–284. South Jeju Island, Korea: Masaryk University.

Takasaki, Toshiyuki & Yumiko Mori. 2007. Design and Development of a Pictogram Communication System for Children Around the World. In Toru Ishida, Susan Fussell & Piek Vossen (eds.), *Proceedings of the First International Workshop on Intercultural Collaboration (IWIC)*, 193–206. Kyoto, Japan: Springer Berlin Heidelberg.

Tambouratzis, George, Marina Vassiliou & Sokratis Sofianopoulos. 2017. *Machine Translation with Minimal Reliance on Parallel Resources*. Switzerland: Springer International Publishing.

Tan, Liling, Jon Dehdari & Josef van Genabith. 2015. An Awkward Disparity between BLEU / RIBES Scores and Human Judgements in Machine Translation. In Toshiaki Nakazawa, Hideya Mino, Isao Goto, Graham Neubig, Sadao Kurohashi & Eiichiro Sumita (eds.), *Proceedings of the 2nd Workshop on Asian Translation (WAT)*, 74–81. Kyoto, Japan.

Tanimoto, Steven L. 1997. Representation and Learnability in Visual Languages for Web-based Interpersonal Communication. In The Institute of Electrical & Electronics Engineers (eds.), *Proceedings of the 1997 IEEE Symposium on Visual Languages (VL)*, 2–9. Capri, Italia: IEEE Computer Society Press.

Tatomir, Iulia. 2003. *Iconic communication*: Delft University of Technology dissertation.

Tee, Kimberly, Karyn Moffatt, Leah Findlater, Eve MacGregor, Joanna McGrenere, Barbara Purves & Sidney S. Fels. 2005. A Visual Recipe Book for Persons with Language Impairments. In Wendy Kellogg, Shumin Zhai, Gerrit van der Veer & Carolyn Gale (eds.), *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, 501–510. Portland, OR: ACM Press.

Tesselaar, Charlotte. 2015. *Tijd en moeilijkheid in pictozinnen*: KU Leuven dissertation.

Thomas, Stephen. 1995. Rhetorical Structure Theory. In Frank Van Eynde & Valerio Allegranza (eds.), *Studies in Machine Translation and Natural Language Processing: Semantics and Discourse: A NLP Perspective*, 159–174. European Commission.

Tiedemann, Jörg. 2009. News from OPUS - A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In Nicolas Nicolov, Kalina Bontcheva, Galia Angelova & Ruslan Miktov (eds.), *Recent Advances in Natural Language Processing, Volume V*, 237–248. Amsterdam, The Netherlands: John Benjamins.

Tiedemann, Jörg. 2012. Parallel Data, Tools and Interfaces in OPUS. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*, 2214–2218. Istanbul, Turkey: European Language Resources Association.

Tijus, Charles, Javier Barcenilla, Brigitte Cambon de Lavalette & Jean-Guy Meunier. 2007. The design, understanding and usage of pictograms. In Denis Alamargot, Patrice Terrier & Jean-Marie Cellier (eds.), *Written Documents in the Workplace*, 17–31. Amsterdam, The Netherlands: Elsevier.

Tjong Kim Sang, Erik, Marcel Bollmann, Remko Boschker, Francisco Casacuberta, Feike Dietz, Stefanie Dipper, Miguel Domingo, Rob van der Goot, Marjo van Koppen, Nikola Ljubesic, Robert Ostling, Florian Petran, Eva Pettersson, Yves Scherrer, Marijn Schraagen, Leen Sevens, Jörg Tiedemann, Tom Vanallemeersch & Kalliopi Zervanou. 2017. The CLIN27 Shared Task: Translating Historical Text to Contemporary Language for Improving Automatic Linguistic Annotation. *Computational Linguistics in the Netherlands Journal* 7. 53–64.

Toutanova, Kristina & Robert C. Moore. 2002. Pronunciation Modeling for Improved Spelling Correction. In Pierre Isabelle (ed.), *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, 144–151. Philadelphia, PA: Association for Computational Linguistics.

Trnka, Keith, John McCaw, Kathleen F. McCoy & Christopher Pennington. 2016. Word Prediction and Communication Rate in AAC. *Telehealth and Assistive Technologies (Telehealth/AT)* 173–176.

Tuset, Pere, Pere Barberán, Léonard Janer, Esteve Buscà, Sandra Delgado & Nuria Vilà. 2010. Messenger Visual: A pictogram-based IM service to improve communications among disabled people. In Ebba Þóra Hvannberg, Marta Kristín Lárusdóttir, Ann Blandford & Jan Gulliksen (eds.), *Proceedings of the 6th Nordic Conference on Human-Computer Interaction (NordiCHI)*, 797–800. Reykjavik, Iceland: ACM Press.

Vaillant, Pascal. 1997. *Interaction entre modalités sémiotiques: de l'icône à la langue*: Université Paris XI Orsay dissertation.

Vaillant, Pascal. 1998. Interpretation of iconic utterances based on contents representation: Semantic analysis in the PVI system. *Natural Language Engineering* 4(1). 17–40.

Vaillant, Pascal & Marie-Françoise Castaing. 2008. Iconicity as Recognizability. *VISIO* 23 pp.

Van Eynde, Frank. 2005. *Part-of-Speech Tagging en Lemmatisering van het D-Coi Corpus*. Leuven, Belgium: Centrum voor Computerlinguïstiek, University of Leuven.

van Noord, Gertjan. 2006. At Last Parsing is Now Operational. In Pet Mertens, Cédrick Fairon, Anne Dister & Patrick Watrin (eds.), *Actes de la 13e Conference sur le Traitement Automatique des Langues Naturelles (TALN)*, 20–42. Leuven, Belgium: Leuven University Press.

van Noord, Gertjan, Gosse Bouma, Frank Van Eynde, Daniël de Kok, Jelmer van der Linde, Ineke Schuurman, Erik Tjong Kim Sang & Vincent Vandeghinste. 2013. Large Scale Syntactic Annotation of Written Dutch: Lassy. In Peter Spyns & Jan Odijk (eds.), *Essential Speech and Language Technology for Dutch*, 147–164. Berlin: Springer International Publishing.

van Rijsbergen, Cornelis Joost. 1975. *Information Retrieval*. London, UK: Butterworths.

Vanallemeersch, Tom & Vincent Vandeghinste. 2015. Assessing linguistically aware fuzzy matching in translation memories. In İlknur Durgar El-Kahlout, Mehmed Özkan, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, Fred Hollowood & Andy Way (eds.), *Proceedings of the 18th Annual Conference of the European Association for Machine Translation (EAMT)*, 153–160. Antalya, Turkey.

Vandeghinste, Vincent. 2002. Lexicon Optimization: Maximizing Lexical Coverage in Speech Recognition through Automated Compounding. In Manuel González Rodríguez & Carmen Paz Suarez Araujo (eds.), *Proceedings of the Third International Confererence on Lan-*

guage Resources and Evaluation (LREC), 1270–1276. Las Palmas, Canary Islands: European Language Resources Association.

Vandeghinste, Vincent. 2004. Shallow Rule-based Parsing. URL: http://www.ccl.kuleuven.be/∼vincent/ccl/SHARPA/ (accessed September, 2018).

Vandeghinste, Vincent. 2008. *A Hybrid Modular Machine Translation System. LoRe-MT: Low Resources Machine Translation. PhD*. Utrecht, The Netherlands: LOT Publications.

Vandeghinste, Vincent. 2012. Bridging the Gap between Pictographs and Natural Language. In Klaus Miesenberger, Andrea Petz & Kerstin Matausch (eds.), *Proceedings of the Research and Development Working Group Online Symposium on Easy-to-Read on the Web*, Research and Development Working Group.

Vandeghinste, Vincent & Yi Pan. 2004. Sentence Compression for Automated Subtitling: A Hybrid Approach. In Stand Szpakowicz & Marie-Francine Moens (eds.), *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, 89–95. Barcelona, Spain: Association for Computational Linguistics.

Vandeghinste, Vincent & Ineke Schuurman. 2014. Linking Pictographs to Synsets: Sclera2Cornetto. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk & Stelios Piperidi (eds.), *Proceedings of the Ninth International Confererence on Language Resources and Evaluation (LREC)*, 3404–3410. Reykjavik, Iceland: European Language Resources Association.

Vandeghinste, Vincent, Ineke Schuurman, Leen Sevens & Frank Van Eynde. 2017. Translating Text into Pictographs. *Natural Language Engineering* 23(2). 217–244.

Vandeghinste, Vincent & Erik Tjong Kim Sang. 2004. Using a Parallel Transcript/Subtitle Corpus for Sentence Compression. In Maria Teresa Lino, Maria Francisca Xavier, Fátima Ferreira, Rute Costa & Raquel Silva (eds.), *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC)*, 231–234. Lisbon, Portugal: European Language Resources Association.

Vickrey, David, Luke Biewald, Marc Teyssier & Daphne Koller. 2005. Word-Sense Disambiguation for Machine Translation. In Raymond J. Mooney (ed.), *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, 771–778. Vancouver, Canada: Association for Computational Linguistics.

Vilar, David, Gregor Leusch, Hermann Ney & Rafael E. Banchs. 2007. Human evaluation of machine translation through binary system comparisons. In Chris Callison-Burch, Philipp

Koehn, Christof Monz & Cameron Shaw Fordyce (eds.), *Proceedings of the Second Workshop on Statistical Machine Translation (StatMT)*, 96–103. Prague, Czech Republic: Association for Computational Linguistics.

Vinyals, Oriol, Alexander Toshev, Samy Bengio & Dumitru Erhan. 2015. Show and Tell: A Neural Image Caption Generator. In Horst Bischof Graz, David Forsyth, Cordelia Schmid & Stan Sclaroff (eds.), *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3156–3164. Boston, MA: IEEE Computer Society Press.

Vossen, Piek, Laura Bloksma & Paul Boersma. 1999. *The Dutch WordNet. URL: http://vossen.info/docs/1999/DutchWordNet.pdf (accessed September, 2018)*. Amsterdam, The Netherlands: University of Amsterdam.

Vossen, Piek, Attila Görög, Rubén Izquierdo & Antal Van den Bosch. 2012. DutchSemCor: Targeting the ideal sense-tagged corpus. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk & Stelios Piperidis (eds.), *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*, 584–589. Istanbul, Turkey: European Language Resources Association.

Vossen, Piek, Isa Maks, Roxane Segers & Hennie van der Vliet. 2008. Integrating lexical units, synsets, and ontology in the Cornetto Database. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis & Daniel Tapias (eds.), *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC)*, 1006–1013. Marrakech, Morocco: European Language Resources Association.

Walmsley, Jan & Kelley Johnson. 2003. *Inclusive Research with People with Learning Disabilities: Past, Present and Futures*. London, UK: Jessica Kingsley Publishers.

Wang, Hsiu Feng, Sheng Hsiung Hung & Ching Chih Liao. 2007. A Survey of Icon Taxonomy Used in the Interface Design. In Willem Paul Brinkman & William Ham, Dong Han Wong (eds.), *Proceedings of the 14th European Conference on Cognitive Ergonomics: Invent! Explore! (ECCE)*, 203–206. London, UK: ACM Press.

Wiegand, Karl & Rupal Patel. 2012. SymbolPath: A Continuous Motion Overlay Module for Icon-based Assistive Communication. In Matt Huenerfauth & Sri Kurniawan (eds.), *Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS)*, 209–210. Boulder, CO: ACM Press.

Wilson, Robert E., Samuel D. Gosling & Lindsay T. Graham. 2012. A Review of Facebook Research in the Social Sciences. *Perspectives on Psychological Science* 7(3). 203–220.

Woodsend, Kristian & Mirella Lapata. 2011. Learning to Simplify Sentences with Quasi-Synchronous Grammar and Integer Programming. In Paola Merlo, Regina Barzilay & Mark Johnson (eds.), *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 409–420. Edinburgh, UK: Association for Computational Linguistics.

Wołk, Krzysztof, Agnieszka Wołk, Krzystof Marasek & Wojciech Glinkowski. 2017. Pictogram-based mobile first medical aid communicator. *Procedia Computer Science* 121. 3–10.

Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel & Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In Sanjoy Dasgupta & David McAllester (eds.), *Proceedings of the International Conference on Machine Learning (ICML)*, 2048–2057. Lille, France: Journal of Machine Learning Research.

Xu, Wei, Courtney Napoles, Ellie Pavlick, Quanze Chen & Chris Callison-Burch. 2016. Optimizing Statistical Machine Translation for Text Simplification. *Transactions of the Association for Computational Linguistics* 4. 401–415.

Yakin, Halina Sendera Mohd. & Andreas Totu. 2014. The Semiotic Perspectives of Peirce and Saussure: A Brief Comparative Study. *Procedia - Social and Behavioral Sciences* 155. 4–8.

Yang, Zhongliang, Yu-Lin Zhang & Sadaqat ur Rheman. 2017. Image Captioning with Object Detection and Localization. In Yao Zhao, Xiangwei Kong & David Taubmann (eds.), *Proceedings of the 9th International Conference on Image and Graphics, Part II (ICIG)*, 109–118. Shanghai, China: Springer International Publishing.

Yazdani, Masoud. 1993. *Multilingual Multimedia: Bridging the Language Barrier with Intelligent Systems*. Bristol, UK: Intellect Books.

Yuizono, Takaya, Jingya Qi & Jun Munemori. 2012. Effects of a Semantic Pamphlet for Pictogram Chat on Small Talk Between Japanese and Chinese Speakers. In Kok-Leong Ong (ed.), *Proceedings of the 7th International Conference on Knowledge, Information and Creativity Support Systems (KICCS)*, 73–80. Melbourne, Australia: IEEE Computer Society Press.

Zhang, Joy Ying & Stephan Vogel. 2006. Suffix array and its applications in empirical natural language processing. Tech. rep. Language Technologies Institute, Carnegie Mellon University Pittsburgh, PA.

Zhou, Zhen. 2014. Research on Pictogram and the Situation of Pictogram Design in China. *Canadian Social Science* 10(6). 162–165.

Zhu, Zhemin, Delphine Bernhard & Iryna Gurevych. 2010. A Monolingual Tree-based Translation Model for Sentence Simplification. In Aravind K. Joshi, Chu-Ren Huang & Dan Jurafsky (eds.), *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*, 1353–1361. Beijing, China: Association for Computational Linguistics.

Dissemination

This section presents an overview of my scientific and public dissemination activities. They are ordered from newest to oldest per category.[1]

Journal Articles

Bulté, Bram, Leen Sevens & Vincent Vandeghinste. In press. Automating Lexical Simplification in Dutch. *Computational Linguistics in the Netherlands Journal* 8.

Sevens, Leen, Vincent Vandeghinste, Ineke Schuurman & Frank Van Eynde. In press. Less is More: A Rule-Based Syntactic Simplification Module for Improved Text-to-Pictograph Translation. *Data & Knowledge Engineering*.

Vandeghinste, Vincent, Leen Sevens & Ineke Schuurman. 2017. E-Including the Illiterate. *IEEE Potentials* 36(1). 29–33.

Vandeghinste, Vincent, Ineke Schuurman, Leen Sevens & Frank Van Eynde. 2017. Translating Text into Pictographs. *Natural Language Engineering* 23(2). 217–244.

Book Chapters

Sevens, Leen, Vincent Vandeghinste, Ineke Schuurman & Frank Van Eynde. 2018. Involving People with an Intellectual Disability in the Development of Pictograph Translation Technologies for Social Media Use. In Louise-Amélie Cougnon, Barbara De Cock & Cédrick Fairon (eds.), *Cahiers du CENTAL, Volume 9*, 57–69. Presses Universitaires de Louvain.

Sevens, Leen, Vincent Vandeghinste, Ineke Schuurman & Frank Van Eynde. 2017. Simplified Text-to-Pictograph Translation for People with Intellectual Disabilities. In Ashwin Ittoo, Le Minh Nguyen & Elisabeth Métais (eds.), *Natural Language Processing and Information*

[1] Dissemination activities that do not directly relate to the work presented in this dissertation are not shown.

Systems: 22nd International Conference on Applications of Natural Language to Information Systems, 185–196. Springer International Publishing.

Conference Proceedings

Saggion, Horacio, Daniel Ferrés, Leen Sevens & Ineke Schuurman. 2017. Able to Read my Mail: An Accessible E-mail Client with Assistive Technology. In Allan Ellis & Tatsuya Hagino (eds.), *Proceedings of the 14th International Web for All Conference (W4A)*, 1–5. Perth, Australia.

Sevens, Leen, Jo Daems, Annelies De Vliegher, Ineke Schuurman, Vincent Vandeghinste & Frank Van Eynde. 2017. Building an Accessible Pictograph Interface for Users with Intellectual Disabilities. In Peter Cudd & Luc de Witte (eds.), *Proceedings of the 2017 Association for the Advancement of Assistive Technology in Europe Congress (AAATE)*, 870–877. Sheffield, UK: IOS Press.

Medina Maestro, Jaime, Horacio Saggion, Ineke Schuurman, Leen Sevens, John O'Flaherty, Annelies De Vliegher & Jo Daems. 2016. Towards Integrating People with Intellectual Disabilities in the Digital World. In Emile Aarts, Hamid Aghajan, Michael Berger, Vic Callaghan, Diane Cook, Sajal Das, Anind Dey, Sylvain Giroux, Pertti Huuskonen, Jadwiga Indulska, Achilles Kameas, Peter Mikulecký, Daniel Shapiro, Toshiyo Tamura & Michael Weber (eds.), *Proceedings of the 7th International Workshop on Intelligent Environments Supporting Healthcare and Well-being (WISHWell)*, 348–357. London, UK.

Sevens, Leen, John J. O' Flaherty, Ineke Schuurman, Vincent Vandeghinste & Frank Van Eynde. 2016. E-Inclusion of Functionally Illiterate Users by the use of Language Technology. In Jan Engelen (ed.), *Proceedings of the 2nd Conference on Engineering4Society*, 91–95. Heverlee, Belgium.

Sevens, Leen, Gilles Jacobs, Vincent Vandeghinste, Ineke Schuurman & Frank Van Eynde. 2016. Improving Text-to-Pictograph Translation Through Word Sense Disambiguation. In Claire Gardent, Raffaella Bernardi & Ivan Titov (eds.), *Proceedings of the 5th Joint Conference on Lexical and Computational Semantics (*SEM)*, 131–135. Berlin, Germany.

Sevens, Leen, Tom Vanallemeersch, Ineke Schuurman, Vincent Vandeghinste & Frank Van Eynde. 2016. Automated Spelling Correction for Dutch Internet Users with Intellectual Disabilities. In Ineke Schuurman, Vincent Vandeghinste & Horacio Saggion (eds.), *Proceedings of 1st LREC Workshop on Improving Social Inclusion using NLP: Tools and Resources*

(ISI-NLP), 11–19. Portorož, Slovenia.

Sevens, Leen, Vincent Vandeghinste, Ineke Schuurman & Frank Van Eynde. 2015. Extending a Dutch Text-to-Pictograph Converter to English and Spanish. In Jan Alexandersson, Ercan Altinsoy, Heidi Christensen, Peter Ljunglöf, François Portet & Frank Rudzicz (eds.), *Proceedings of 6th Workshop on Speech and Language Processing for Assistive Technologies (SLPAT)*, 110–117. Dresden, Germany.

Sevens, Leen, Vincent Vandeghinste, Ineke Schuurman & Frank Van Eynde. 2015. Natural Language Generation from Pictographs. In Anya Belz, Albert Gatt, François Portet & Matthew Purver (eds.), *Proceedings of 15th European Workshop on Natural Language Generation (ENLG)*, 71–75. Brighton, UK.

Sevens, Leen, Vincent Vandeghinste & Frank Van Eynde. 2014. Improving the Precision of Synset Links Between Cornetto and Princeton WordNet. In Jorge Baptista, Pushpak Bhattacharyya, Christiane Fellbaum, Mikel Forcada, Chu-Ren Huang, Svetla Koeva, Cvetana Krstev & Eric Laporte (eds.), *Proceedings of the Workshop on Lexical and Grammatical Resources for Language Processing (LG-LP)*, 120–126. Dublin, Ireland.

Conference Abstracts

Vandeghinste, Vincent, Leen Sevens & Ineke Schuurman. 2018. Pictograph Translation Technologies for People with Limited Literacy. CLARIN Annual Conference. Pisa, Italy, 8-10 October 2018.

Bulté, Bram, Leen Sevens & Vincent Vandeghinste. 2018. Automating Lexical Simplification in Dutch. Meeting of Computational Linguistics in The Netherlands - CLIN28. Nijmegen, The Netherlands, 26 January 2018.

Sevens, Leen, Vincent Vandeghinste, Lyan Verwimp, Ineke Schuurman, Patrick Wambacq & Frank Van Eynde. 2018. Pictograph-to-Text Translation for Augmented and Alternative Communication. Meeting of Computational Linguistics in The Netherlands - CLIN28. Nijmegen, The Netherlands, 26 January 2018.

Sevens, Leen, Ineke Schuurman, Annelies De Vliegher & Jo Daems. 2017. Unity is Strength: How User Feedback Influenced Technical Decisions in the Able to Include Project. NNDR 14th Research Conference. Örebro, Sweden, 3-5 May 2017.

Sevens, Leen, Vincent Vandeghinste, Ineke Schuurman & Frank Van Eynde. 2017. Make it Simple, But Significant: Improved Text-to-Pictograph Translation for People with Intellectual Disabilities. NNDR 14th Research Conference. Örebro, Sweden, 3-5 May 2017.

Sevens, Leen, Vincent Vandeghinste, Ineke Schuurman & Frank Van Eynde. 2017. Syntactic Simplification for Improved Text-to-Pictograph Translation. Meeting of Computational Linguistics in The Netherlands - CLIN27. Leuven, Belgium, 10 February 2017.

Sevens, Leen, Ineke Schuurman, John O'Flaherty, Vincent Vandeghinste & Frank Van Eynde. 2016. Pictographs to the Rescue! Social Media for Functionally Illiterate Users. Lancaster Disability Studies Conference. Lancaster, UK, 6-8 September 2016.

Sevens, Leen, Ineke Schuurman, Vincent Vandeghinste & Frank Van Eynde. 2016. Pictograph Translation Technologies for Improving E-Inclusion. PLIN 2016. Louvain-la-Neuve, Belgium, 12 May 2016.

Jacobs, Gilles, Leen Sevens, Vincent Vandeghinste, Ineke Schuurman & Frank Van Eynde. 2015. Word Sense Disambiguation in Text-to-Pictograph Translation. Meeting of Computational Linguistics in The Netherlands - CLIN26. Amsterdam, The Netherlands, 18 December 2015.

Sevens, Leen, Vincent Vandeghinste, Ineke Schuurman & Frank Van Eynde. 2015. Natural Language Generation from Pictographs. Meeting of Computational Linguistics in The Netherlands - CLIN26. Amsterdam, The Netherlands, 18 December 2015.

Sevens, Leen, Ineke Schuurman, Vincent Vandeghinste & Frank Van Eynde. 2015. Automatic Translation with Pictographs to Serve People with IDD. NNDR 13th Research Conference. Bryggen, Norway, 8 May 2015.

Sevens, Leen, Vincent Vandeghinste, Ineke Schuurman & Frank Van Eynde. 2015. Text-To-Pictograph Translation for Six Language Pairs. Meeting of Computational Linguistics in The Netherlands - CLIN25. Antwerp, Belgium, 6 February 2015.

Awards

- Mediawijs: M-Award (September 2017, Brussels) for K-point (Thomas More Geel) and the Centre for Computational Linguistics for their contributions to the Able to Include Project.

- Pioniersprijs Humane Wetenschappen 2017: Honourable Mention (May 2017, KU Leuven, Leuven) for Leen Sevens, Vincent Vandeghinste, Ineke Schuurman, and Frank Van Eynde for their work on the pictograph translation technologies.
- Best Paper Award (April 2017, Web For All Conference, Perth) for Horacio Saggion, Daniel Ferrés, Leen Sevens, and Ineke Schuurman: "Able to Read my Mail: An Accessible E-mail Client with Assistive Technology".
- Falling Walls Lab: Audience Award & Jury's 1st Prize (March 2017, KU Leuven, Leuven) for Leen Sevens: "Breaking the Wall of Illiteracy".
- Language Industry Award "Best Language Service 2015" (March 2016, De Taalsector, Brussels) for the Text-to-Pictograph translation technology.
- Best Poster Award (June 2015, LOT Summer School, Leuven) for Leen Sevens, Vincent Vandeghinste, Ineke Schuurman, and Frank Van Eynde: "Text-To-Pictograph Translation for Six Language Pairs".

Samenvatting

Nooit eerder werd er in onze maatschappij meer tekst geproduceerd dan op de dag van vandaag. De opkomst van sociale media-websites, zoals Facebook of Twitter, speelt hierin een niet mis te verstane rol. Wie niet (goed) kan lezen of schrijven, loopt het risico om uitgesloten te worden. Geschreven tekst kan een barrière vormen tussen mensen met en zonder een verstandelijke beperking.

Gespecialiseerde websites, zoals het veilige internetplatform WAI-NOT, proberen tegemoet te komen aan de noden van mensen met lees- en schrijfproblemen door middel van tekst-naar-spraaktechnologie, helder taalgebruik en educatieve spelletjes. WAI-NOT ontwikkelde daarnaast een simpele methode voor automatische conversie van tekst naar pictogrammen. Pictogrammen worden gebruikt in het bijzonder onderwijs, dagcentra en beschutte werkplaatsen om de communicatie tussen mensen met een verstandelijke beperking en hun omgeving te bevorderen. De vertaalmethode van WAI-NOT bleek echter onbruikbaar; een woord werd slechts geconverteerd wanneer er een exacte overeenkomst gevonden werd tussen dat woord en de bestandsnaam van een pictogram. Niet alleen bleven zo de meeste woorden onvertaald, er werden op deze manier ook veel fouten geïntroduceerd. Het voorzetsel *bij* werd bijvoorbeeld steevast geconverteerd naar *bij.png*, het pictogram van het insect.

Vandeghinste & Schuurman (2014) ontwikkelden een intelligenter vertaalsysteem. Ze koppelden duizenden Sclera- en Beta-pictogrammen, twee sets die gebruikt worden in Vlaamse scholen en dagcentra, aan synsets in Cornetto (Vossen et al. 2008). Cornetto is een lexicaal-semantische database die meer dan 119.000 Nederlandse lemma's omvat, verspreid over meer dan 84.000 synsets. Synsets zijn groepen van woorden met dezelfde betekenis, zoals *kapot* en *stuk*. Door pictogrammen te koppelen aan synsets, konden Vandeghinste & Schuurman het lexicale bereik van hun vertaaltechnologie vergroten. Bovendien konden zij op deze manier ook gebruikmaken van de semantische relaties tussen synsets, zoals de antonymie- of hyperonymierelaties.

Het baselinesysteem voor Text-to-Pictograph-vertaling van Vandeghinste et al. (2017) maakt gebruik van basistechnieken uit natuurlijke taalverwerking, waaronder automatische woordsoortannotatie en lemmatisering (Hoofdstuk 3). Inputtekst wordt vrij letterlijk vertaald; zo is er bijvoorbeeld geen verandering in woordvolgorde en worden enkel de lidwoorden verwijderd. We evalueerden het baselinesysteem aan de hand van automatische metrieken, manuele beoordelingen en observaties en focusgroepen met mensen met een verstandelijke beperking, en voerden drie aanpassingen door.

De eerste aanpassing betreft het ontwikkelen van een systeem voor automatische spellingscorrectie van tekst geschreven door mensen met een verstandelijke beperking (Hoofdstuk 4). Foutief gespelde inputtekst resulteert vaak in incorrecte of incomplete pictovertalingen. Vanwege het gebrek aan parallelle (authentieke - gecorrigeerde) tekst, ontwikkelden we een methodologie die slechts een kleine hoeveelheid aan manueel gecorrigeerde tekst vereist. Het toepassen van herschrijfregels op sequenties van lettertekens staat in deze methode centraal. Deze herschrijfregels werden automatisch afgeleid uit een klein parallel corpus van berichten geschreven door mensen met een beperking en de manueel verbeterde versie. Wanneer alle spellingsvarianten voor een foutief gespeld woord gegenereerd zijn, wordt de meest waarschijnlijke combinatie van varianten gekozen aan de hand van lettertekengebaseerde fuzzy matching. We toonden aan dat deze benadering tot contextgevoelige spellingscorrectie tot betere resultaten leidt dan de meer conventionele technieken, zoals methodes die gebruikmaken van taalmodellen.

De tweede verbetering is een module voor syntactische simplificatie (en compressie), die ervoor zorgt dat inputtekst automatisch vereenvoudigd wordt voordat deze naar pictogrammen vertaald wordt (Hoofdstuk 5). De bedoeling van deze module is om de consistentie, helderheid en leesbaarheid in pictovertalingen te bevorderen - dit in tegenstelling tot de vrij letterlijke pictogram-per-woord-vertalingen van het baselinesysteem. Door het gebrek aan parallelle (authentieke - vereenvoudigde) tekst, ontwikkelden we een regelgebaseerde methode die gebruikmaakt van automatische syntactische zinsontleding. Per bericht is slechts één syntactische ontleding nodig, zelfs indien er zich meerdere te vereenvoudigen fenomenen voordoen. Aan de hand van automatische metrieken, manuele beoordelingen en gebruikerstests met leerkrachten, begeleiders en familieleden, konden we aantonen dat vereenvoudigde pictosequenties doorgaans als “minder moeilijk” beschouwd worden of vaker correct geïnterpreteerd worden dan niet-vereenvoudigde pictosequenties. We ontwikkelden daarnaast eveneens een module voor het automatisch vereenvoudigen van werkwoordsgroepen en temporaliteitsdetectie, die vlotte pictolezers toelaat om de temporele kenmerken van een reeks pictogrammen te achterhalen.

De derde en laatste aanpassing betreft de implementatie van een tool voor automatische desambiguering van woordbetekenissen in de Text-to-Pictograph-pipeline, die ons toelaat om waarschijnlijkheidsscores aan synsets te koppelen (Hoofdstuk 6). Tijdens het berekenen van een optimale pictovertaling wordt de selectie van het meest plausibele pictogram bepaald door de synset met de hoogste score. In tegenstelling tot het baselinesysteem, dat aanvankelijk de meest frequente betekenis van een woord selecteerde, is het nieuwe systeem voortaan minder geneigd om een foutief pictogram te kiezen voor een ambigu woord.

We evalueerden de verbeterde Text-to-Pictograph-pipeline met ervaren pictolezers en concludeerden dat het nieuwe systeem een verbetering is ten opzichte van het baselinesysteem, zowel in termen van accuraatheid, als in termen van adequaatheid (Hoofdstuk 7).

Daarnaast ontwikkelden we ook een systeem dat pictogrammen automatisch omzet naar natuurlijke tekst. Aangezien het Pictograph-to-Text-systeem van pictografische input (of een combinatie van geschreven tekst en pictogrammen) gebruikmaakt, ontwikkelden we een statische pictohiërarchie die gebruikers toelaat om categorieën te doorbladeren en pictoberichtjes op te stellen (Hoofdstuk 8). Aan de basis van de interface ligt een groot corpus van door gebruikers gegenereerde content, dat we gebruikten om taalkundige kennis af te leiden. Op deze manier konden we een hiërarchie ontwikkelen die aangepast is aan de woordenschat en interesses van de eindgebruikers. Tijdens de hands-on sessies konden we concluderen dat de nieuwe hiërarchie, die volgens de principes van user-centered design ontwikkeld werd, een sterke verbetering is ten opzichte van het oude categorisatiesysteem van WAI-NOT. Naast de statische picto-interface, ontwikkelden we eveneens een dynamische pictovoorspeller, die op basis van eerder geselecteerde pictogrammen contextueel relevante of semantisch gelijkende pictogrammen aan de eindgebruiker suggereert.

Voor de eigenlijke vertaling van pictogrammen naar natuurlijke tekst evalueerden we verschillende benaderingen (Hoofdstuk 9). De baselinesystemen maken gebruik van 5-gram- en long short-term memory-taalmodellen. We toonden aan dat 5-gram-gebaseerde benaderingen tot betere resultaten leiden dan long short-term memory-gebaseerde benaderingen, en dat het gebruik van kleinere corpora die het taalgebruik van de eindgebruikers op gepaste wijze modelleren te verkiezen is boven het gebruik van grote, gemixte corpora. In onze statistische machinevertalingsgebaseerde benaderingen tot Pictograph-to-Text-vertaling vergeleken we phrase-based methodes met neurale machinevertalingstechnieken. Aan de hand van automatische metrieken stelden we significante verbeteringen vast ten opzichte van de baselinesystemen, meer specifiek bij het gebruik van neurale technieken. Deze conclusie werd gestaafd door adequaatheids- en vlotheidsbeoordelingen en paarsgewijze ranking door menselijke annotatoren.

Met de ontwikkeling van Text-to-Pictograph- en Pictograph-to-Text-vertaaltechnologieën dragen we bij tot de inclusie van mensen met een verstandelijke beperking in online omgevingen. De technologieën kunnen vlot uitgebreid worden naar andere talen (Hoofdstuk 10) en de verschillende subcomponenten van de tools kunnen ingezet worden voor tal van andere toepassingen, waaronder het ontwikkelen van apps voor tweedetaalverwerving of hulpmiddelen voor mensen met dyslexie of afasie.