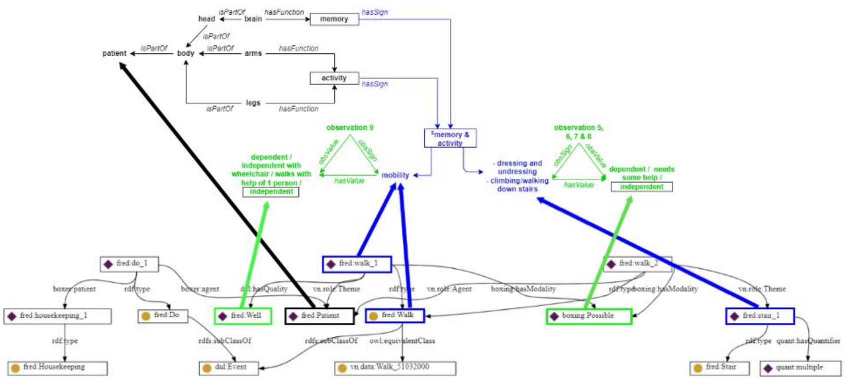
Universiteit Utrecht

Care2Report: Dialogue summarisation for geriatric performance assessments

Author

Koen van den Braak



First supervisor:
Prof. Dr. S. Brinkkemper

Second supervisor:
Dr. Fabiano Dalpiaz

Master’s programme Business Informatics (MBI) Thesis

January 2023

Contents

1. Introduction

A consequence of improved medical care is the requirement of having to document a large quantity of information. Not only is it vital for medical professionals to have access to a patient's medical history, but also for insurance purposes and, in some instances, the law. In nursing, a large portion of work consists of updating the client's electronic medical record (EMR). Nursing is a sector in healthcare which focusses on human-environment health as a whole and healing through caring (Smith, 2019). Considering that nursing is a broad concept, in this thesis the focus will primarily be on nursing homes and home care. Nursing homes are defined as facilities that provide day-and-night support and care in a home-styled environment for vulnerable people with complex health issues (Orrell, Tolson, Abbatecola et al., 2015). There are over 400,000 employees active at nearly 1,000 nursing institutions in the Netherlands (CBS 2021). The sector has a combined revenue of over 18 billion euros. The EMR is the electronic patient's medical history, which is updated by the care provider over time and includes all relevant information for a person's care (Centers for Medicare & Medicaid Services, 2012). The EMR includes medication, observations, treatments and medical history. In addition, for nursing specifically it contains health goals, information handover to colleagues, caregiver information and incidents. The EMR allows professionals in healthcare to make informed decisions based on the available data. Well implemented nursing EMR systems can have a positive effect on communication, handover, care efficiency and workflow (Hardiker, Dowding, Dykes & Sermeus, 2019). Currently, the EMR is largely updated manually by care providers.

Manually inputting information into an EMR can be time-consuming. It is an additional task that is often not adequately integrated with the nurse's primary responsibilities. Meanwhile, advancements are made in the field of speech technology. While manual input is still favoured for some tasks, voice based systems are a logical choice for tasks that involve conversations to begin with.. Speech technology is a compound term for various technologies that either automatically recognize speech or have a voice-based interface. Automatic speech recognition (ASR) is a process in which a speech signal is transformed to a word sequence through in a computer program implemented algorithm (Huang, Acero & Son). ASR has been a concept for multiple decades now, but only with recent technology has it been possible to implement it in day-to-day human-machine communication (HMC) and human-human communication (HHC) (Yu & Deng, 2016). This increase in value is partly due to the increase in computational power, but also due to a vast increase in available data that algorithms can use to train. ASR can be a useful method for dictation, data entry, voice search and personal digital assistance (PDA) like Siri on Iphone (Yu & Deng, 2016). Speech technology as a method for inputting medical information is useful, but EMRs are more than a transcription of medical consultations.

Natural language processing (NLP) is a subfield of computer science which allows computers to access human language. According to Liddy (2001) it is considered an AI discipline, among other disciplines, as the goal of NLP is to achieve human-like performance. It is a set of methods that is ingrained in technology in everyone's life. It allows mailboxes to recognize spam based on the language used and it allows search engines to come up with accurate search results based on the user input (Eisenstein, 2019). NLP is often achieved by machine learning, which is effective because of the availability and relative simplicity of textual data. It does however have its drawbacks, because although sentences can syntactically be correct, that does not mean it semantically makes sense. NLP is primarily used for textual analysis. However, combining it with speech technology allows speech (audio) to be used as input. Speech technology and NLP will be discussed in more detail.

1.1 Problem statement

EMR offers better informed decision making for care providers. However, there are multiple problems with the current implementation of EMR in nursing. Studies have shown that it is hard to balance structured information (medicine and treatments) with more nuanced information like patient insight (Hardiker et al., 2019) (Groot et al., 2017). During medical consults, the care provider usually updates the EMR during the consult. In nursing this often happens after the fact. This has implications, because nurses might forget certain important information when their shift is nearly over.

Professionals in nursing express dealing with a high regulatory pressure. A survey among district nursing professionals in the Netherlands reveals that 25% of their time is spent on administration (KPMG, 2020). This is a slight improvement compared to 2019, where 27% of the time was spent on administration. However, professionals have indicated that 15% would be an acceptable number, meaning that the difference of 10% would rather be spend on quality care than on administration. A different source reported in 2019 that the experienced administrative burden was as high as 35%, while 23% was regarded as an acceptable number (Berenschot, 2019). Costs for an administrative burden of 31% were estimated to be €5 billion (Berenschot, 2018), while a decrease to 14% would only cost half. While the EMR increases the quality of care, it leaves less time for the client.

The current administrative burden is not the only problem. The population of the Netherlands is ageing rapidly. According to the Dutch central office for statistics (CBS), 18.5% of the population was over the age of 65 in 2017 (CBS, 2021). In 2021 this percentage increased to 20.0%. The CBS predicts that in ten years this number will further increase to 23.8%. In absolute numbers this means an increase from 3.5 million elderly in 2021 to 4.2 million in 2030. An increase in elderly people will result in a higher demand for nursing professionals, as well as nursing homes. Failure to meet these demands will result in a shortage of professionals and an even higher workload. An alternative is to find a solution to the administrative burden of the current workforce, thereby allowing them to have more time for clients and quality healthcare.

1.2 Research objective and research questions

The aim of this thesis is to design a solution to the administrative burden in the nursing care sector and geriatric sector by designing and validating a dialogue summarization and interpretation pipeline. This will be a modification of the existing Care2Report (C2R) pipeline. As of this moment, C2R has been explored for a limited number of healthcare disciplines, primarily for a general practitioner on the topic of otitis externa (swimmer's ear). The conversation interpretation algorithm is therefore optimised for this specific domain. Different domains however have different types of protocols, assessments and conversations. To truly unburden the healthcare sector, there is a need for research in more domains. The findings of this study may be generalised to other domains.

Based on the research objective, the following main research question has been defined.

MRQ: Which linguistic techniques can be used in a pipeline as a solution to the administrative burden in geriatric performance assessment and nursing?

To design an algorithmic solution that is able to tackle the problem of the administrative burden in geriatrics and nursing, an understanding has to be built of the problem. Furthermore, it is required to recognize how geriatric assessments are performed and how that knowledge can be transferred to an algorithm. To provide an answer to the main question, the following five sub questions were derived from the main question.

RQ1: What administrative issues do the nursing care and geriatrics sectors face?

Research question 1 provides an extensive understanding of the problem by performing a literature review. This is a prerequisite to identifying the stakeholders and requirements necessary to find a solution. In addition to the literature review, an interview will further identify the main administrative problems the nursing sector faces.

RQ2: What is the state-of-the-art in speech technology, NLP and AI within the healthcare domain?

Research question 2 will discuss the current state and uses of speech technology and NLP. Furthermore, an overview will be given of the use of NLP and AI specifically in healthcare and nursing. The drawbacks of choosing various NLP and ARS methods as a solution will be discussed as well.

RQ3: How are geriatric conversations structured?

A previous C2R study collaborated with the Radboud UMC Nijmegen to retrieve recordings of multiple comprehensive geriatric assessments (Kemper, Brinkkemper & Dalpiaz, 2021). This research questions examines conventions and patterns in geriatric conversations between doctors and patients..

RQ4: How can a consultation transcript be matched with geriatric ontologies?

A Medical Guideline Ontolgy was already created specifically for geriatric conversations (Kemper, Brinkkemper & Dalpiaz, 2021). This research question examines how the ontology can be linked to the transcripts, which is necessary for reporting. It will examine a method to process transcripts in such a way that they can be linked to a geriatric ontology. To achieve this, we will design a set of transformations that allows us to extract the narrative out of dialogues.

RQ5: How to automatically extract narratives from geriatric dialogues?

In the final research question 5 we investigate a method to automate the transcript processing mentioned in RQ4. This processing is required for the C2R pipeline to interpret the data. The goal of this research question is to use syntactic elements, such as predicates, objects, and subjects, extracted from dialogue sentences, to automate the narrative information extraction transformations described in RQ4.

1.3 Thesis Outline

This thesis is structured in the following way. Chapter 2 will explain the research method, including the use of the design cycle and the method of literature review. The following chapters (chapters 3 to 7) will each answer one of the five sub questions. Chapter 3 and 4 (RQ 1 and 2) are mainly composed of literature reviews. Chapter 2 will discuss de administrative burden in nursery. Then, chapter 3 will discuss the state-of-the-art in NLP and ASR. Chapter 5 discusses patterns and conventions found in geriatric performance assessments. Chapter 6 will describe the way geriatric consultation transcripts can be matched to ontologies. Chapter 7 discusses the automatization of summarising geriatric dialogues and finally in chapter 8 the project will be discussed and a conclusion is formed.

2. Research method

2.1 Design cycle

This research project uses the Design Cycle method (Wieringa, 2014). Design science researches and designs artifacts in a problem context for the purpose of improving that context. Examples of said artifacts are algorithms, methods and techniques, while the context for design science is usually the development and maintenance of software and information systems. Design science differentiates between design problems and knowledge questions. Design problems result in the design of an artifact that change the real world, which solution may differ based on the researcher, problem context and stakeholders. Knowledge questions aim to find a single right answer to the question. This research designs an algorithm (artifact) in the context of the administrative burden in nursing and geriatrics (problem context) and is therefore a design problem. Wieringa's design cycle identifies three tasks (figure 2.1). The design cycle is part of the engineering cycle, which also includes a treatment implementation task. However, given that implementation is not the outcome of this study, the design cycle is used. Figure 2.1 originally included the implementation task, but it has been removed for clarity. **Problem investigation** consists of researching the problem context and understanding what and why the context needs improving. **Treatment design** is the design of the artifact that can improve the context. **Treatment validation** is the task of validating the artifact and identifying if the artifact is a proper solution to the problem. Each research question can be linked to a phase of the design cycle. RQ1 - 3 are identified as problem investigation, RQ4 and 5 as treatment design and RQ5 includes the validation task as well. (figure 2.2).

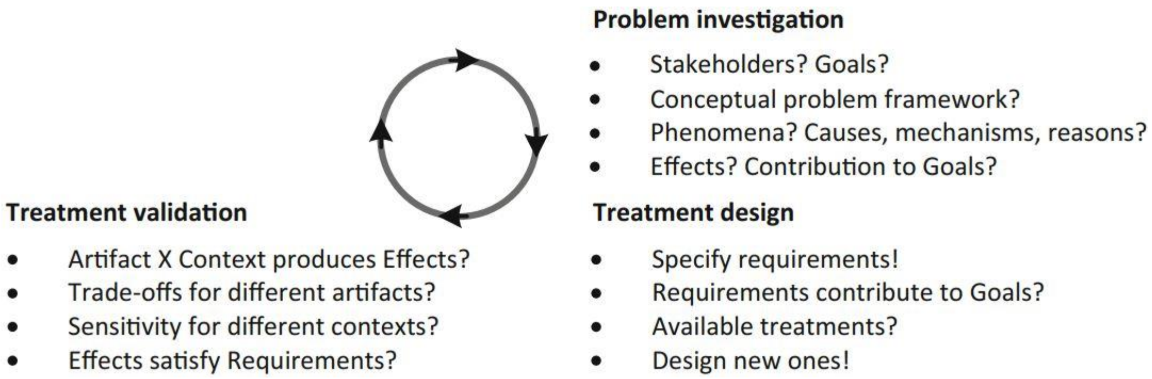


Figure 2.1: Design cycle (Wieringa, 2014)

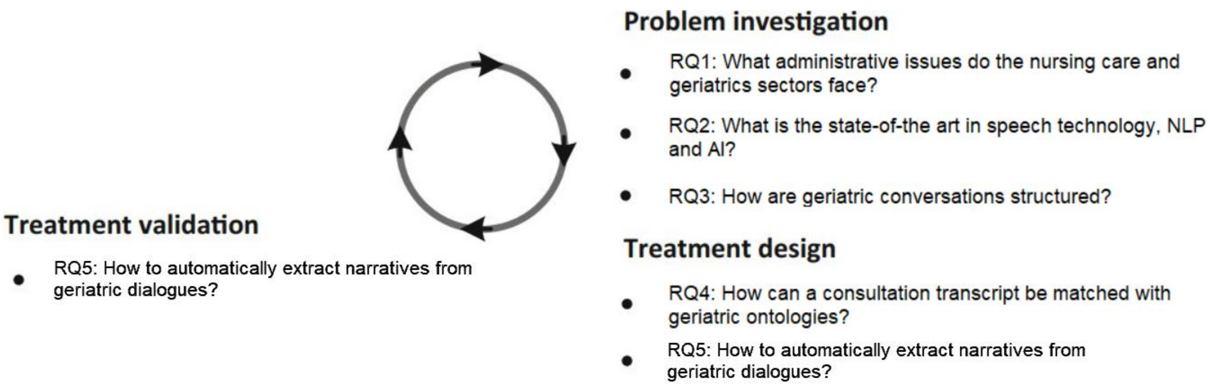


Figure 2.2: Research questions in the Design cycle (Wieringa, 2014)

2.2 Problem investigation

2.2.1 Literature study

Problem investigation is done using three different methods: a literature study, interviews and a case study. Figure 2.3 depicts the various step undertaken to gain an understanding of the design problem. As the figure shows, a literature study will provide answers primarily for RQ1 and 2. Often a distinction is made between systematic literature review, semi-systematic literature review (narrative review) and an integrative review (Snyder, 2019). Keeping in mind the research questions and scope of the research area, a protocol is chosen which mostly resembles the semi-systematic literature review. For research question 1 the goal is to get a comprehensive understanding of the administrative burden in nursing. This includes inquiring knowledge about the stakeholders, causes and effects of the problem. However, due to the different stakeholders, research groups and angles it will be difficult to perform a systematic literature review. The goal is not to find empirical evidence, but rather to get a full overview of the problem. The same reasoning applies to RQ2. Rather than finding the empirical evidence to a question, we want to get a complete picture of the state-of-the-art.

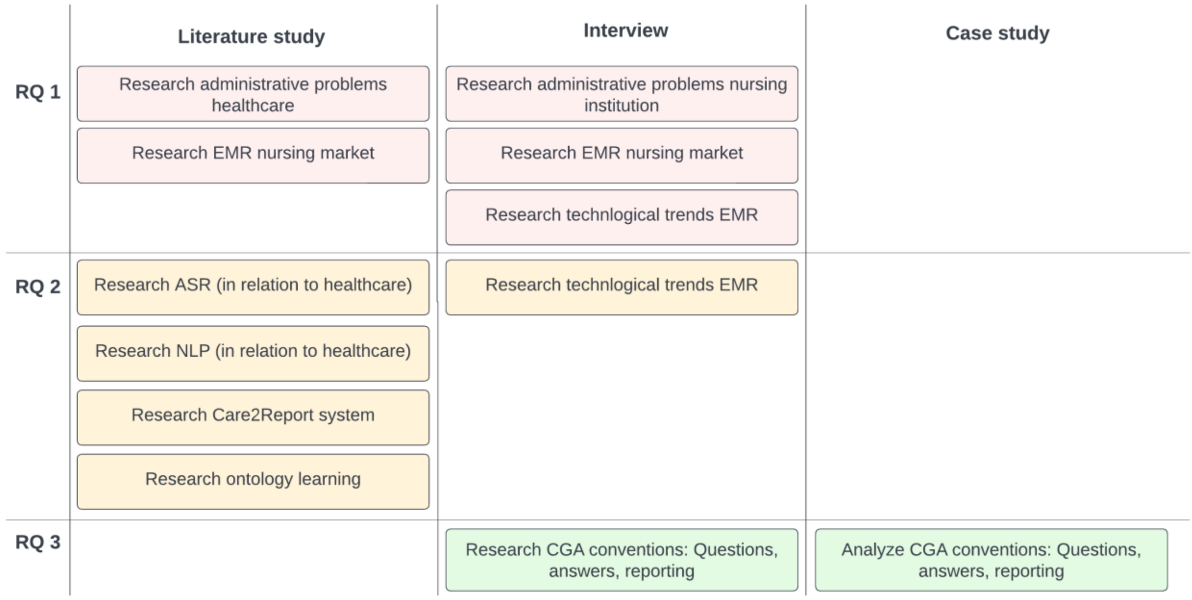


Figure 2.3: Problem investigation activities

Although the review is not fully systematic, it is still possible to apply rules and systems to the review. Table 2.1 shows the criteria used for the literature review of RQ1 and 2, including: keywords, search engines, publication year and sources for each sub-question. Both Dutch and English keywords were used. Other languages were excluded for this literature review. Rarely were the terms in Table 1 directly searched for; rather, queries were utilized to
get to the most relevant papers. For example: rather than just entering EMR, *"EMR" AND "Administrative burden"* were entered. For research question 1, a recent publication year is chosen as it is more important to review recent literature in order to have an understanding of current administrative problems instead of past problems. Initially 2017 is set as a lower limit to prevent now irrelevant reports. Recency is important for research question 2 also, as we research the state-of-the-art in NLP techniques. Older research can still be relevant today however, and publication year is therefore not a hard constraint. RQ2 will also include a detailed explanation of the C2R pipeline. The method for gaining an understanding of this pipeline is by means of published C2R papers.

Table 2.1: Semi-systematic literature review – RQ 1 & 2

	RQ1	**RQ2**
Keywords*	EHR, EMR, Electronic health record, Nursing, Administrative burden, District nursing, home care, Geriatrics, Healthcare	Automatic speech recognition, ASR, Speech technology, NLP, Natural language processing, Ontology learning, Care2Report
Search engine	Google Scholar, Google (grey literature)	Google Scholar
Publication year**	2017 - now	2017 – now
Sources	Journals, books, conferences, statistical office, nursing institution reports	Journals, books, conferences, Care2Report publications

**Dutch translations of these words were used as well. Does not include all derivative words*
*** There are a few exceptions*

After the initial keywords are found, snowballing is used to expand to more specific areas of the subject. Backward snowballing is the process of using the reference list to find new papers that the main article cites. Forward snowballing is the process of looking at the papers that cited the main article (Wohlin, 2014). The review employs both types of snowballing, but primarily backward snowballing.

Reports on nursing, and in particular the administrative burden, often come from healthcare knowledge institutions such as Nivel and consulting firms such as M&I/Partners. In contrast to other fields, a large share of the information is not available in scientific research, but rather in grey literature. To ensure the quality of this information, grey literature will be used for sources that are respected and consulted by nursing institutions and the government.

2.2.2 Interview

For research question 1, an interview with the chief information officer (CIO) of a major elderly care institution is conducted to provide more information about the administrative burden and EMR market in the Dutch nursing sector. The interview identifies what requirements care institutions have with regards to EMR systems. It will also discuss the potential of using ASR for EMR systems. The interview will be referenced to multiple times in chapter 3.

Furthermore, a meeting took place with prof. dr. Yvonne Schoon of the geriatrics department at Radboud UMC Nijmegen's to learn more about how CGAs (comprehensive geriatric assessments) are carried out. During this conversation, the analysed transcripts will be discussed to find conventions in the execution and reporting of CGA's. This information will be used for RQ 3 and will later be applied to the design of the conversation interpretation algorithm.

2.2.3 Case study

Within the broad ambition of Care2Report to find a solution to the administrative burden in healthcare as a whole, this thesis focusses on nursing and geriatrics. Therefore a case study will be done for geriatrics specifically. To find all components for an algorithmic solution able to interpret geriatric conversations, first an understanding must be build of how geriatric conversations are performed and structured. The most widespread method of performing a geriatric assessment is the comprehensive geriatric assessment (CGA). This method will be extensively explained in chapter 5 (RQ 3). Data collection has already been performed by students Kendall Kemper and Rick Oostveen, alumni from Utrecht University. Kemper collaborated with the geriatrics department at the Radboud UMC Nijmegen to record audio of CGA sessions of thirteen patients. Kemper then transcribed those recordings and kept twelve of the transcripts for her study (Kemper, Brinkkemper & Dalpiaz, 2021). For privacy, all personal information which could be traced to the patient have been removed such as names, street names and other named entities. In chapter 5, we will analyse those transcripts and find conventions and patterns in CGA that can be used in the design of a rule-based algorithm. Furthermore, the RQ3 analysis will build on earlier analysis performed by Rick Oostveen, who structured the transcriptions in Excel and identified question types and answer types.

2.3 Treatment design

RQ4 and RQ5 involve the treatment design phase of the design cycle. The treatment will be designed by altering the existing C2R pipeline. Specifically, by altering the way medical conversations are summarised and interpreted. The knowledge acquired in RQ1-3 will be used to automatically summarize geriatric conversations, and in the long run allow it to interpret geriatric conversations and generate CGA scores automatically which can be inserted in the EMR directly. During the design process, choices that can be applied to healthcare sectors other than geriatrics are considered. The treatment design can be divided into two parts: theoretical treatments and automated treatments.

RQ4 primarily involves the design of a theoretical treatment. To initiate the treatment design, a model of the C2R pipeline is altered to highlight the key differences between the general and the geriatric C2R approach. From there, the various processes relating to the geriatric ontology and knowledge graph matching are discussed. A combination of standard C2R techniques and new techniques are used and validated to try and find a recommended pipeline for the purpose of automated geriatric assessment. An approach to transforming geriatric dialogues to narratives with the aim of improving triple generation is developed during RQ4.

RQ5 involves automating the narrative information extraction that was theorised in RQ4. A set of linguistic techniques are deployed to automatically extract the most important information out of dialogues, so that processes further in the pipeline can be streamlined. This approach makes use of dependency parser SpaCy and sentence realiser SimpleNLG. The method of designing this treatment consists of multiple cycles in which the output of the treatment will serve as a new baseline to improve the design.

2.4 Treatment validation

Treatment validation has the objective of testing whether the proposed treatment of RQ4 and RQ5 is a potential solution to the investigated problem in the context of this project. The primary method of validation is by comparing the output of the designed treatment to the required or preferred output. The treatment can be accepted if the narrative output contains all information from the input dialogue that is required for a geriatric assessment, without the obstacles caused by working with multiple speaking turns. To compare the output to the required output, a system is used in which the output is given a rating out of the following three: incorrect, partially correct and correct. By analysing the performance of the design based on these three ratings, and discussing possible causes for the results, a conclusion can be reached on whether the suggested treatment is valid.

3. Administrative burden in nursing

The problem statement summarised the administrative burden in nursing, as well as the effect of the aging population on the problem. This section will discuss the problem and potential solutions in more detail. A large elderly nursing institute's chief information officer (CIO) was interviewed to gain insight into the administrative burden, nursing landscape, and IT solutions. Their institute is currently looking for a new EMR, and he shared their key considerations in selecting a new system. He will be referred to in this section multiple times.

My initial research would focus on designing a solution to the administrative burden in the nursing sector. It would focus on finding requirements for a to be designed solution that would either be integrated with existing EMR providers or would exist as a separate entity. However, due to personal preferences and difficulties this research will now focus on geriatrics. There are similarities between both sectors and so as to not discard any relevant research, the administrative burden in nursing will still be explored. The findings of this thesis may be applicable to the field of nursing.

3.1 Administrative burden

In 2017 in the Netherlands, home care nurses worked an average of 13.2 hours on administrative tasks a week. In nursing homes this is an average of 11.2 hours and about two thirds of this time is client related. Client related administrative tasks often consist of updating the EMR, while non-client administration are often juridical tasks or tasks necessary to declare costs. Almost 66% of professionals in home care and nursing homes express that the administrative burden is too high (de Veer, de Groot, Brinkman, Francke et al., 2017). The same study shows that only 44% of nurses find the IT system for administration clear and easy to work with. Based on this report, Nivel (Dutch institute for healthcare research) identified four possible solutions.

1. Management has to take responsibility for reducing the administrative burden by avoiding double registrations and the registration of unimportant information.
2. Find a way to integrate administrative tasks in regular work. Again, managers have a key role here.
3. Find better and easy to use IT solutions.
4. Decrease the number of hours spend on administration.

Regarding the first option, the CIO stated that a survey conducted in his organization revealed that employees have too much to report. The Netherlands has a health-care system with many regulations. This is required to declare expenditures, making it unfeasible to simply reduce the amount of reports. Double registration is a concern not just internally, but also externally. If a patient is transferred from a hospital to a nursing home, both parties have to manually enter that information into their respective systems. Looking at the third solution in Nivel's report, the CIO revealed that fax machines are still very common in nursing institutions. Old EMR systems frequently have a difficult-to-use interface. EMR providers are aware of these issues and have integrated features and updated their systems to address some of them. An extensive overview of the EMR market will be discussed later in this section. The fourth approach may appear to be the most straightforward, but it is the most difficult to implement in practice.

3.2 Dutch EMR market

The Dutch EMR market has a large number of competitors, but there are three main providers according to the CIO: PinkRoccade, Nedap and Ecare. Market research by M&I/Partners (Eurlings & Schaik, 2020), a Dutch IT consultancy firm, reveals that the two EMR's with the largest market share are

PinkRoccade (24%) and Nedap (37%). The next largest competitor is Cura (12%). The number of EMR providers has decreased over the last few years, and may decrease further in the upcoming years due to consolidation. A few smaller EMR providers have merged into larger ones. However, they are still relatively small compared to Nedap and PinkRoccade. An overview of the EMR providers in nursing homes can be seen in figure 3.1.

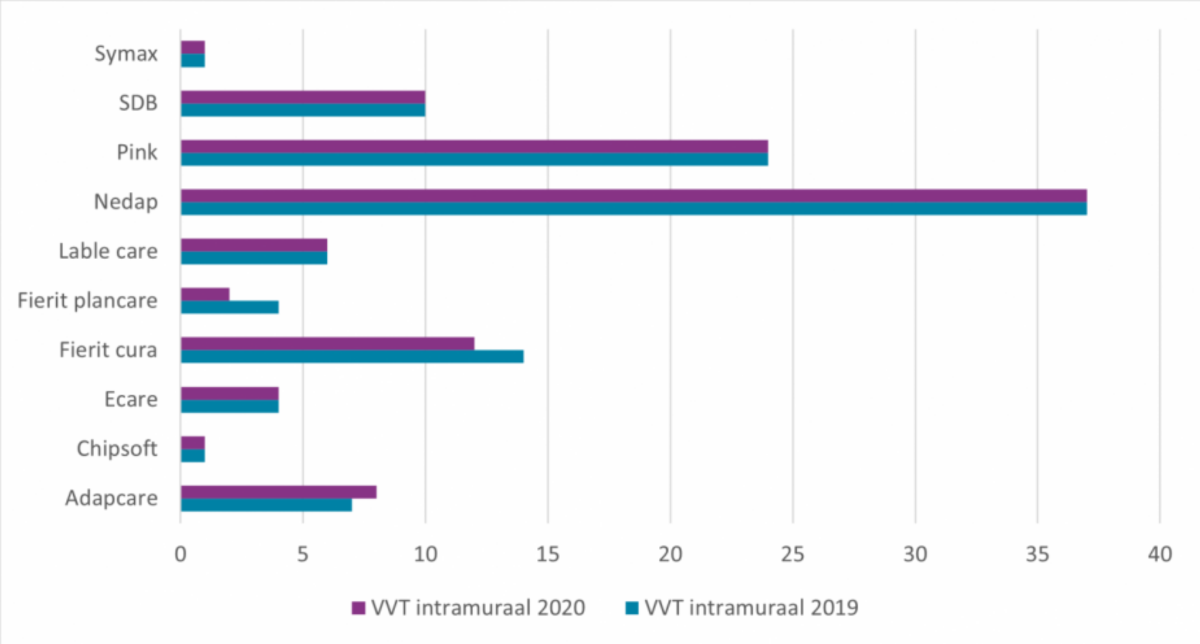


Figure 3.1: EMR Market, percentage market share intermural nursing in the Netherlands (Eurlings & Schaik, 2020)

A large development over recent years is the addition of client portals, allowing for better communication between client and carer. Clients and family are able to log in to the client portal to see communication between nurses and between nurses and clients. Another development in the market is the use of SaaS-solutions (software as a service). According to the interviewed CIO, nursing institutions value this highly, given that they often do not have the experience and time to integrate their systems with their current infrastructure. It has to operate practically immediately. This may explain the success of Nedap, as it is one of their main selling points. The other key points for the CIO were first and foremost that it can demonstrable improve the working and administrative digital environment for care professionals, as long as it is within reasonable budget. The second point is that the EMR is flexible and open for new Dutch and international developments. If developments happen on a technological or juridical level the EMR should be able to adapt to those developments. The third point is that it uses modern technology, including it being SaaS. There should be low maintenance for the organisation. The final point is if it integrates with the goals and architecture of the organisation. It should include agendas, have a simple user experience etc. Many of the largest EMR providers now take into account administration by making the applications more mobile based. This means that the EMR's can often be run on phones or tablets, removing the requirement of sitting behind a computer or laptop.

3.3 Standardised reporting

To reduce the administrative burden, communication between healthcare institutions has to improve. Due to the substantial IT landscape, it seems unfeasible to get one homogenous architecture, but progress can be made in the way these organisations communicate. These organisations currently communicate largely with unstructured email, phone calls and faxes. According to the CIO, there are initiatives in the Netherlands for a standardised way of reporting.

A frontrunner in the Netherlands is Nictiz with an initiative called eOverdracht (eTransfer). This initiative standardises the way clients can be signed up for a transfer. Nictiz makes use of zorginformatiebouwstenen (zib), or health data building blocks in English (Nictiz, 2021a). These are standardised information elements that can be used in multiple healthcare information systems. The zib are primarily based on SNOMED CT, which is the international standard for medical terminology. A zib contains a main concept, which is linked to data identified by a standardized ID. An example of the patient zib can be found in figure 3.2.

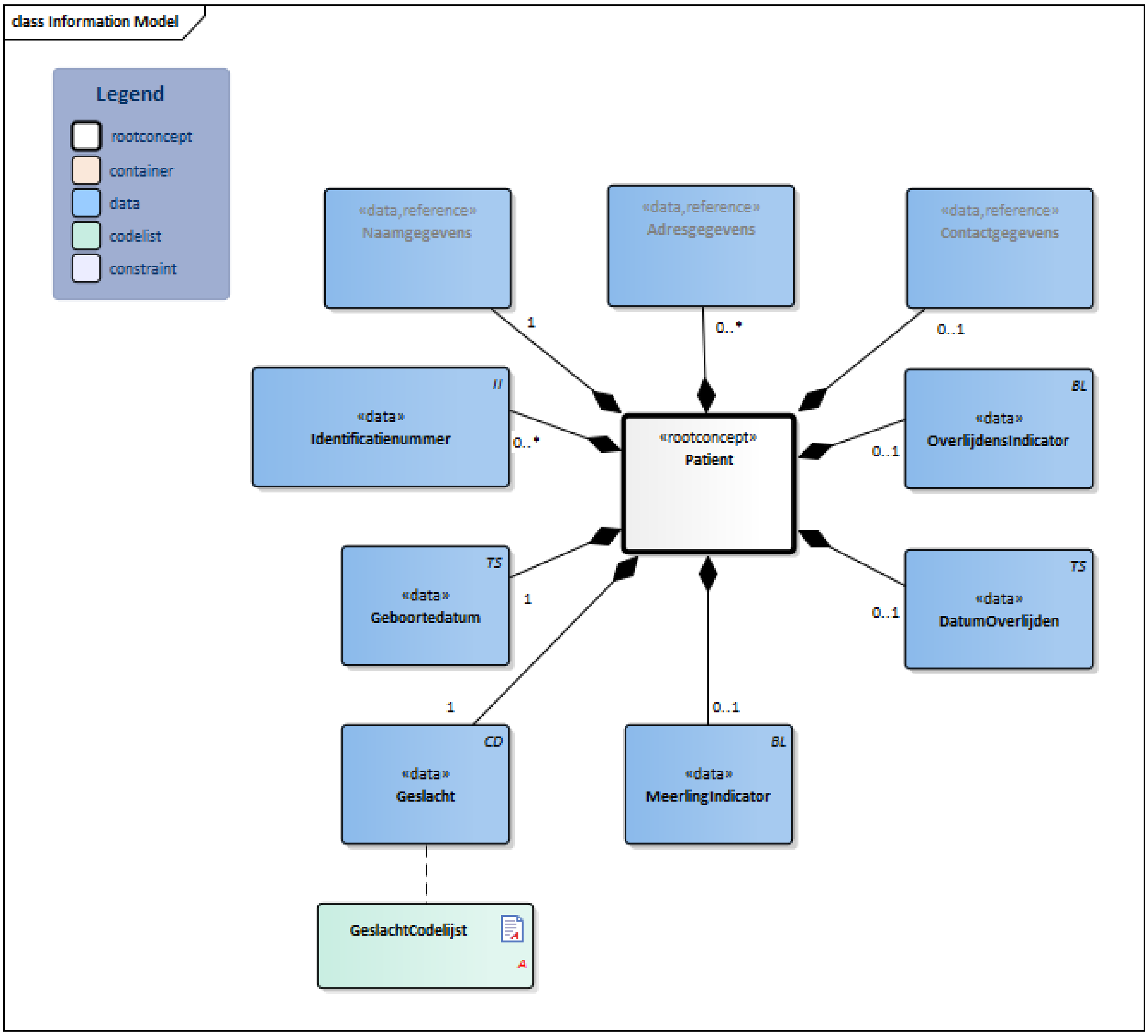


Figure 3.2: zorginformatiebouwstenen (zib) of Patient. class Information model of data concepts (zibs.nl, 2020)

The client transfer in nursing should contain the following five elements if it is to be compatible with eOverdracht (Nictiz, 2021b). Note that not all examples of data elements are always required.

1. *Administrative information:* including personal data and the date of transfer.
2. *General patient context:* family, social, juridical context
3. *Medical context:* practitioner, diagnoses, allergies, measured values
4. *Nursing context – care plan:* Nurse, health goals, current patient problems
5. *Nursing context – specification health status:* selfcare, mobility, diet

According to Nictiz, eOverdracht is used “in many places”, although it is unknown how many exactly. This is however only one standard, and many more could be implemented to make a difference on the administrative burden.

4. Speech technology and NLP in healthcare

4.1 Automated speech recognition

To propose a solution to the administrative problem in nursing and geriatrics, it is important to understand the state-of-the-art in speech technology, NLP and Care2Report and how it might decrease time spent on administration. This section will solely focus on these technologies within the scope of the healthcare domain. ASR can be used for a variety of applications: dictation, PDAs and voice command searches, but while a voice controlled system could be useful, it is not helpful during medical consultations because it would distract from the conversation. An EMR contains a summarised report of a consultation, storing only the most relevant information. For that reason, using speech to text alone to transcribe a full consultation would be insufficient. However, while ASR alone is not the solution to the problem, it is the starting point. To automatically summarise a conversation, the first step is to make sure that audio is transformed to text, so that the system eventually is able to read and transform it.

ASR systems typically consist of four components: Signal processing and feature extraction, acoustic model, language model and hypothesis search (Yu & Deng, 2016). These four components are depicted in Figure 4.1. The data input of an ASR system is an audio signal. The signal processing component removes noise from the audio signal and transforms the audio signal into audio vector features. These features are various descriptions of sound formatted in a machine readable data type. This transformation to vector features makes the audio readable by an acoustic model. An acoustic model is a representation of human speech. Each phoneme or character in human speech correlates to an audio wave or vector feature and constructs the acoustic model. The vector features are compared to the acoustic model and generates an AM score to find the most likely sounds. To get a more accurate result however, a language model is involved. A language model takes into account which sounds and words are predicted to follow each other given the grammatical constraints of the language. The language model generates a LM score. The hypothesis search will then combine the AM score and LM score and generate the highest probability result.

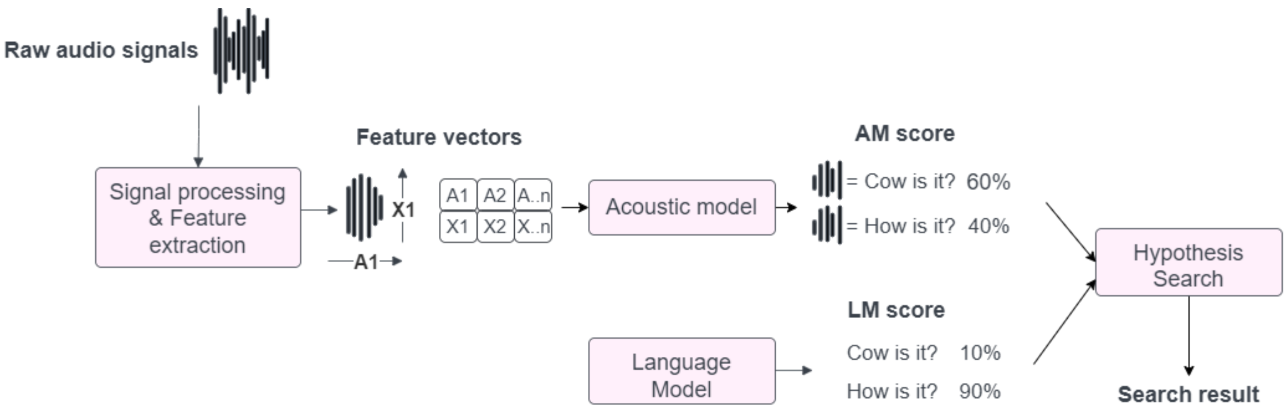


Figure 4.1: standard ASR components

ASR performance

Given the above described process, the quality of the output is dependent on the chosen feature extraction method, the acoustic model and language model. These models are trained using training corpora. The results generally improve if the language model has more knowledge of the selected domain (Yu & Deng, 2016). For example, if a conversation takes place about certain muscles, and a doctor uses the Latin medical terms, the ASR system would output better results if the model is trained on a corpus with medical terms.

Considering that different models can yield different results, there are ways to calculate performance. One effective way to measure the performance of ASR systems is the Word Error Rate (WER). To calculate this metric, the correct transcript is compared to the output of the ASR. The Word Error Rate is calculated by adding the number of substitutions (S), deletions (D) and insertions (I) and dividing it by the total number of words spoken (N) (Ali & Renals, 2018). A substitution is a word that was replaced by a different word. A deletion indicates words that were left out by the ASR and insertions are words not present in the original audio. The lower the WER the better the performance, because a low WER indicates that the ASR output contained less errors compared to the correct transcript.

$$WER = \frac{S + I + D}{N}$$

Although the WER is a good way to measure performance, it does not say anything about the reason for the performance. The WER can change based on background noise, microphone quality and pronunciation. One major issue in ASR, which may cause issues for nursing and geriatrics in particular, is the fact that ASR does not work equally well for all people. Research shows that the WER for older adults (60+) is higher than those of younger adults (20-60) (Werner, Huang & Pitts, 2019). Even the best performing tests on older adults contained more errors than those of younger adults. The lower ranges range from 3.7% (younger adults) to 14.2% (80+). The research also showed however that results can be improved by training the model on a larger number of people. For example, the upper-value WER was 46.8% for a model trained on less than 400 people, while it was 29.1 for a model trained on more than 400 people. Not only age affects WER, but also gender and race. According to a literature review of multiple studies, in most cases, the WER of a female sample was lower, except for cases where the language model was trained on a predominantly male sample. One study found a WER of 0.19 for white speakers compared to 0.35 for black speakers (Koenecke et al., 2020). However, it did note that there were not only racial differences, but also regional differences between the test subjects which may have influenced the WER.

While the WER is often the most important metric in choosing and adjusting ASR systems, for many applications the real time factor (RTF) is as important. Whereas WER measures accuracy, RTF measures speed. The RTF is calculated by dividing the system's processing time by the duration of the audio. For example, if one would talk to a system for 2 minutes and it would take 10 minutes to see the text on screen, the RTF is 5. For real-time applications a RTF of 1 or close to 1 would therefore be ideal, as the speech recognition would run simultaneously with the audio. While the RTF does not have to be exactly 1 per se for reporting medical consultations, a low RTF is preferred because the entire system would be a pipeline of several phases, each requiring processing time.

$$RTF = \frac{Response\ time}{Speech\ duration}$$

Speech to text comparison

Several criteria must be examined while selecting the best text to speech API for use in a healthcare reporting application. First and foremost, the WER should be low to prevent errors. As geriatrics and nursing are concerned with the elderly, the WER should be particularly low for these groups. Second, the RTF should be reasonably low so that a report may be completed as soon as consultations conclude. Third, the language model should be trained on medical terms so that the ASR system can recognise phrases that may not be used in everyday conversation. Secondary requirements are ease of implementation, use, costs and supported languages. Given these constraints, some of the biggest and most elaborate open- and closed-sourced text to speech APIs will be compared to find the best theoretical candidate for healthcare reporting application. These candidates are Google's Speech-to-Text, Kaldi and IBM Watson.

The first and most well-known API discussed here is Google's Speech-to-Text web API. Speech-to-Text is based on deep learning neural networks and supports around 120 languages. Speech-to-Text includes multiple pre-trained models to transcribe audio files which can be selected to match the user's source by specifying the required model in a config file (Google, 2022). Two of these models are medical dictation (medical professional) and medical conversation (between medical professional and patient). The first 60 minutes can be tried for free after which it costs around $0.96 to $1.44 per hour. Google Speech-to-Text is a cloud service and therefore requires an internet connection and Google Cloud account. It is hard to estimate an exact WER, because that all depends on the request and language, but in general it is seen as having one of the lowest error rates of all ASR systems.

Kaldi is an open source free ASR toolkit written in C++ (Povey et al., 2011). The goal was to provide an easy to use yet modifiable code. Kaldi is seen as one of the fastest and most accurate open-source speech recognition tools and with modern algorithms (Matarneh et al., 2017). It has less premade acoustic models than Google, but it is possible to make and train its own. One study found Kaldi's RTF to be three to four times slower on average than Speech-to-Text, however both values were around 1 or lower making real time processing possible (Kimura, 2018). The WER for Google's API was lower on average, although it is impossible to have similar settings for open-source and closed-source ASR systems, making comparisons difficult.

IBM Watson speech to text includes pre-trained language models of around 13 languages, including English and Dutch. It uses machine learning to continuously improve the combination of acoustic and language models. The tool is able to detect noise while also trying to reduce it. The free version allows for 500 minutes of transcription per month using 36 pre-trained models. For around 1 dollar an hour, the premium version allows customization and training of models. One study found the WER to be slightly worse compared to Google's toolkit for casual conversations, but it is unknown how it would perform for medical consultations (Filippidou & Moussiades, 2020). IBM Watson speech to text can be integrated using multiple languages and seems to have a less robust architecture than Google.

The easiest and most accurate option to use is still Google's Speech-to-Text. An extra positive is that it includes trained models for healthcare. The drawbacks are that it is not free, requires internet connection and syncing with a Google cloud account which may lead to privacy issues. Still, for early developing and testing Google's speech-to-text is probably the best option. A good alternative is IBM Watson, which although slightly less accurate, may be easier and safer to integrate. If Care2Report would like to optimize the speech recognition even further than it could have to train it's own acoustic and language model, making Kaldi a more viable option.

4.2 NLP

Natural languages are essentially unstructured information. Naturally, this means that it is difficult for computers to process compared to structured data. The range of techniques used to find, represent and generate natural language is called natural language processing (NLP). The idea of automatically processing natural language is not new, in fact as early as after world war 2 experiments were done with the automatic translation of Russian to English (Jones, 2001). After that, notable American linguist Chomsky laid an important foundation for NLP with the introduction of generative grammar, which is a theory that grammar can be seen as a set of rules or statements that when applied can form all possible sentences (Halle, 1962). Regarding grammar as a logical structure simplifies the automated generating of sentences. Nowadays, the field of NLP is thriving due to its applicability in a wide variety of research fields, including healthcare. Choosing the appropriate NLP technique depends on the goal and domain. Many techniques exist but there is no one size fits all. Due to rapid developments in the field, one preferred method of doing things may change in the future.

There are two main approaches to NLP: rule-based algorithms and machine learning (ML). Both approaches have their uses depending on the domain and goals and neither is definitively better than the other. A combination of rule-based and ML systems in many cases will yield the best results. ML techniques are generally less human-labour intensive and require less domain knowledge, while rule-based systems are more transparent and are particularly useful for tasks with clear defined structures and rules. Lets say the goal of a task is to find all names in a body of text – also called named entity recognition – and replace them with the word “patient”. A rule-based approach could be to find all words starting with a capital letter. This separate rule would not be sufficient, as the starting words of a sentence would also be marked as a named entity. More complex rules would need to be added to yield accurate results and deal with all possible exceptions. In the case of named entity recognition, rule-based approaches are a thing of the past due to the rule complexity, and are outperformed by statistical ML or hybrid approaches (Mohit, 2014). The appeal of ML approaches, requiring less human effort while simultaneously becoming more advanced, seems like the obvious choice for many tasks. However, ML approaches require extensive training data and are therefore often less suitable for highly specific problems. Although ML is less human labour intensive, it costs time and resources to find and train a sufficient large dataset.

SimpleNLG

Summarizing dialogues can be achieved using various tools. Extractive summaries use words or sentences from the original dialogue, while abstractive summaries generate original sentences based on the most important information in the dialogue. At some point a linguistic generator is required to generate abstractive summaries.

SimpleNLG is a tool initially developed by Ehud Reiter and Albert Gatt, which can generate syntactically correct sentences based on a few parameters (Gatt & Reiter, 2007). It is a Java API realization engine, able to create human language based on a set of lexical and phrasal features and values. It is therefore useful for generating syntactically simplistic sentences in which the preferred format of the output sentence is known. Due to its simplistic and robust nature and well documented transparent API, it is often used as a component in user interfaces or as a research and teaching tool. Although released in 2007, the tool has been developed and updated since. The official release supports English only, but researchers have released libraries for different languages, as well as a port for Python. This tool will be used for research question 5 as a way of automatically summarizing geriatric conversations.

4.3 NLP and speech technology in healthcare

Opportunities

Speech-based technology can be a valuable tool in healthcare. It can save time, as well as remove the need for a professional to be present at all times. Studies highlight some of the possible applications of speech technology, but also highlight some of the challenges speech applications face (Latif et al., 2020). Speech technology can enhance the speech of people with speech disorders. ASR systems can learn to recognize expressions or faults in a person's speech and if necessary correct or improve the pronunciation (Saz et al., 2009). Second, professionals can use speech interfaces to improve work efficiency. This thesis focuses on geriatrics, but trials were done in the past in a clinical environment which provided evidence for an increase in document processing speed, as well as a decrease in time spent on emails (Vogel et al., 2015). Third, speech technology appears to be a useful way to not only recognize emotions, but also mental and psychological conditions.

Challenges

While speech-based technology offers a lot of opportunities for healthcare, it also comes with challenges. In healthcare data is often sensitive. Vulnerabilities in ARS systems could lead to the storage of incorrect data which could further lead to incorrect decision making. ASR systems that rely on deep learning risk being the target of adversarial attacks. Adversarial attacks are attacks that try to either alter input data or input wrong data to ultimately change the output of the model. One study showed that an automatic speech recognition model can be altered with relative ease using adversarial attacks (Carlini & Wagner, 2018). When dealing with ASR systems in healthcare, security and privacy are vital. Medical information is not allowed to leak out. Therefore, ASR system users should be safe against undesired use.

A second challenge is the scarcity of speech data, particularly for languages other than English. For a language model to properly work, it needs to learn from a large amount of data. This is often not a problem for English, but there is a large number of languages that are spoken by fewer people, which makes gathering enough sample data difficult. Healthcare deals with people from all demographics, not just middle-aged adults with perfect pronunciation. It is hard to find a corpus familiar with all medical terms and speech.

There are cases, although scarce, in which faults in ASR systems led to medical malpractice. The impact of these cases ranged from insignificance to death. One study found 9 cases of medical malpractice due to ASR faults (Topaz et al., 2018). For example, in one case a patient got ten times the required dose because of a dictation error. It should be noted that in none of these cases the ASR was the only problem. Better awareness or checks by clinicians could have prevented these cases. Although the chance of high impact medical malpractice is probably lower for nursing than in hospitals, it is still something to take into consideration. It is therefore important that the nurse always checks the output of an ASR system for correctness.

4.4 Ontology learning

To understand how C2R represents data and generates reports, it is important to know what ontology learning is. An ontology in computer science is a formalized specification of concepts and relations in a conceptualization (Gruber, 1995). Using a common ontology results in consistency between agents, and allows data to be processed by machines. Whenever authors write texts, they have a certain conceptual model in their mind. The creation of ontologies (ontology learning) can be in fact compared to reverse engineering the author's written text back to the model in the author's mind (Asim et al., 2018). This can be done manually, which is accurate but time-consuming, but an increasing amount of studies explores the possibilities of automating this process, including for healthcare. A full ontology consists of multiple layers: terms, concepts, relations and axioms. To extract these layers, various linguistic and statistical techniques can be used, becoming more complex as the layers progress. (Asim et al., 2018). It is for example relatively simple to extract terms and synonyms out of text, but it is harder to automatically find relations between terms and create a hierarchy of these terms and relations.

Linguistic methods make use of the properties of languages, and are primarily used for the first layers of ontology learning, such as the extraction of terms and concepts and relations. The first linguistic techniques often used are part-of-speech tagging (POS) and parsing. POS takes words and punctuation (tokens) and assigns word classes to those tokens. Parsing usually results in a parse tree in which the relation between all tokens are shown. To give an example of POS and parsing we take the following sentence: *A symptom of otitis externa is ear pain.* Tagging the tokens in this sentence would give the following word classes (Table 4.1):

Table 4.1: Part-of-speech tags of the sentence: "A symptom of otitis externa is ear pain"

Word class	**Abrv.**	**Tokens**
verb	v	is
determiner	d	A
preposition	prep	of
pronoun	pron	I, my
noun	n	symptom, otitis, externa, ear, pain

After that, a parser is run to extract and visualise the dependency between the words and word groups. SpaCy is used for this example, but many alternatives exist. In the visualised parse in figure 4.3 we can see the grammatical dependencies between words. For example, SpaCy's parser identified that 'ear pain' and 'otitis externa' are compounds (multi-word expressions). The verb 'is' has symptom as a subject, the preposition 'of' is related to the object 'otitis externa' etc.

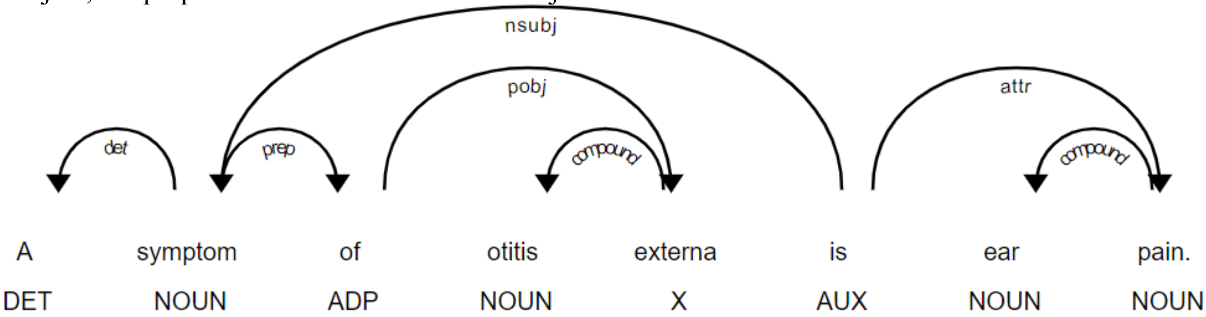


Figure 4.3: parse tree example - SpaCy

These techniques are a good start, as naming the word sequences and relations allows for further processing, but the addition of statistical methods is needed to extract the important concepts out of the corpus. Statistical methods often involve machine learning and aim to use probabilities and other statistics to find relevant terms and concepts, as well as deal with complex sentence structures and multi-word terminology (Asim et al., 2018). Contrastive analysis is one statistical technique, that is used to filter out irrelevant terms from the source. One way to do this is by statistically measuring the domain relevance by using one domain relevant corpus and one non-relevant corpus (Navigli & Velardi, 2002). Other techniques include clustering, which identifies groupings of similar words, and co-occurrence analysis, which seeks to identify words or word groups that frequently occur together.

4.5 Care2Report

Care2Report (C2R) is an extensive ongoing program which involves many different projects and studies executed by PhDs and students of all levels at Utrecht University. It aims to reduce the administrative burden in healthcare by speech and action recognition during consultation sessions (Maas et al., 2020),. Care2report uses speech and action recognition to automatically generate reports in four phases (figure 4.4).

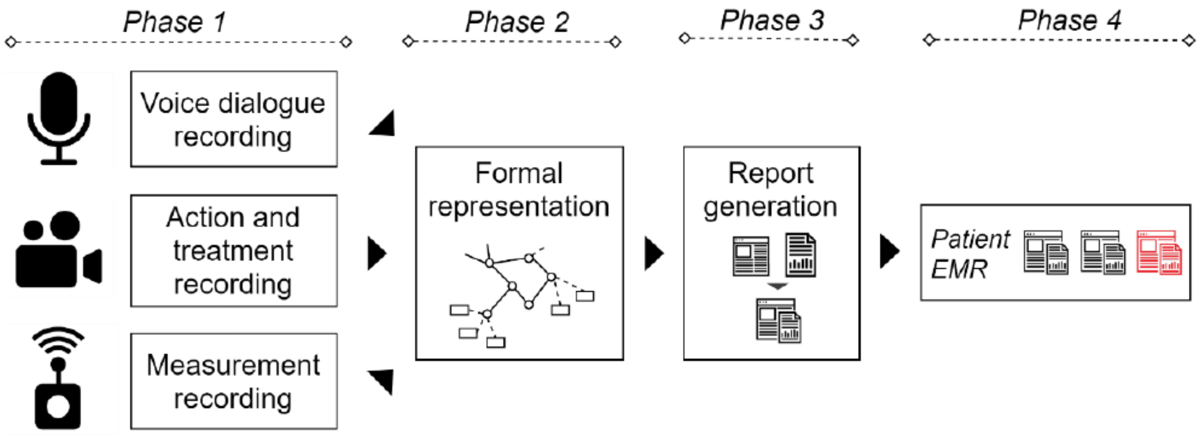


Figure 4.4: Care2Report's 4 phases

Phase 1 (Recording): The first phase consists of both audio recognition and video recognition using a microphone, camera and sensors. Although Care2Report is able to record and recognize visual actions and measurements, this thesis focusses on voice recording only.

Phase 2 (Interpretation): Raw audio and video is transformed into usable data. The audio is transcribed using speech-to-text technology. In the current architecture this is done using Google Cloud Speech-to-text. Currently most software used in the C2R architecture is off the shelf. Python-frog (for Dutch) and FRED (English) are then used to recognize concepts and relations between those concepts in the text. These concepts will form semantic triples in the form of Subjects > Predicates > Objects. An example of a triplet is {:patient, :diagnosedWith, :Influenza}. These triples combined form a knowledge graph (KG). Besides the knowledge graph, a medical guideline ontology (MGO) is created using Protégé. In the next step the ontology is populated with the triples, meaning that the ontology and the triples are linked using a rule-based algorithm, thereby creating a patient medical graph (PMG). This conversation interpretation phase is the focus of the study and will be explained in more detail in the following sections.

Phase 3 (Report generation): NaturalOWL is used to generate natural language by extracting the most important information out of the patient medical graph.

Phase 4 (Patient EMR): The generated report is uploaded to the Patient EMR.

When mentioning C2R to the interviewed CIO, he was receptive towards a voice-based automated reporting solution. Linking this back to the Nivel's four possible solutions to the high administrative workload, a voice-based system would affect the following three solutions. First, it would integrate administrative work in regular work by recording conversations. This process would run simultaneously to the assessment or conversation, thereby integrating it into routine work. Second, given that it is more natural for people to use their voices than to interact with graphical user interfaces, the system should be straightforward to use. Of course, this highly depends on the quality of implementation. Finally, it should decrease the time spend on administration by skipping the step of manually typing information in the EMR.

4.5.1 Medical Guideline Ontology

ElAssy and contributing C2R researchers developed a semi-automatic method to create machine-processable domain-specific ontologies representing medical guidelines called medical guideline ontologies (MGO) (ElAssy et al., 2022). The MGO is part of the ontological conversation interpretation pipeline (figure 4.5). The sources for the MGO are both medical guidelines and SNOMED CT. Medical guidelines are defined as digital documents that describe and define procedural instructions for anamnesis, diagnosis and treatment in healthcare services for the benefit of improved patient's health and wellbeing, care quality and medical decision making. These medical guidelines are often published by both international and national health authorities and are the standard for diagnosing and treating patients. The MGO should therefore accurately reflect the entire guideline and include all possible symptom and treatment options (ElAssy et al., 2022).

SNOMED CT is the world's most comprehensive standardised collection of medical terms. Encoding medical terms allows for more effective and accurate clinical documentation (SNOMED, 2022). While SNOMED CT has an extensive terminology, in its current form it can not be used to represent medical guidelines and thereby medical conversations well. The proposed solution is to map out medical guidelines with the used terminology of SNOMED CT to find all relevant concepts.

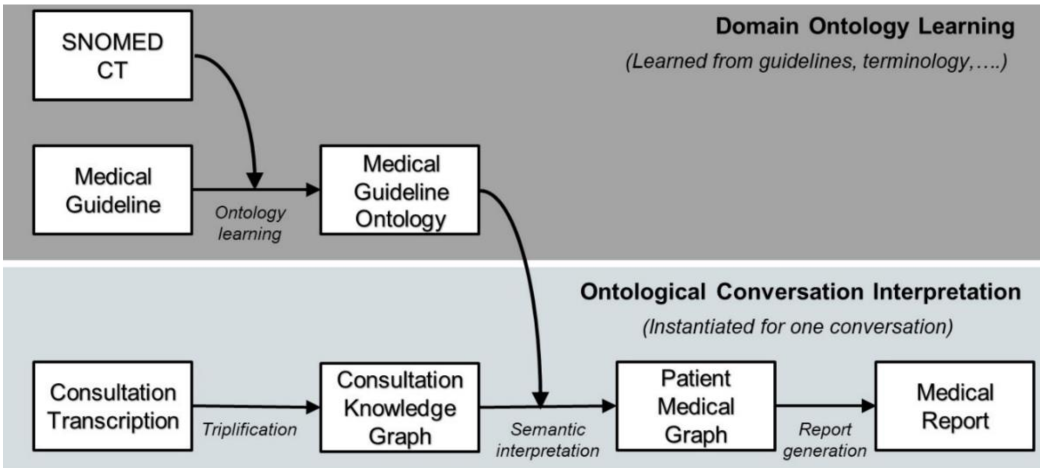


Figure 4.5: C2R ontological conversation interpretation pipeline (ElAssy et al., 2022)

The medical guideline ontology (MGO) consists of five subontologies. the Patient Anatomy Ontology (PAO) represents the anatomy of functions and anatomical structures. The Patient Symptoms Ontology (PSO) includes all the possible patient symptoms relating to the medical guideline. The Patient Observations Ontology (POO) includes all relevant observations the medical professional makes about the patient. The Patient Diagnosis Ontology (PDO) represents the diagnoses of the patient's conditions. Finally, the Patient Treatment Ontology (PTO) represents the professional's prescribed treatment. These five subontologies together form the MGO and are related to each other as depicted in figure 4.6 These five ontologies follow the Dutch standard of reporting called SOEP: Subjective (symptom), Objective (observation), Evaluation (diagnosis) and Plan (treatment). The fifth subontology (PAO) defines the main structure of the ontology and connects to the observations and symptoms.

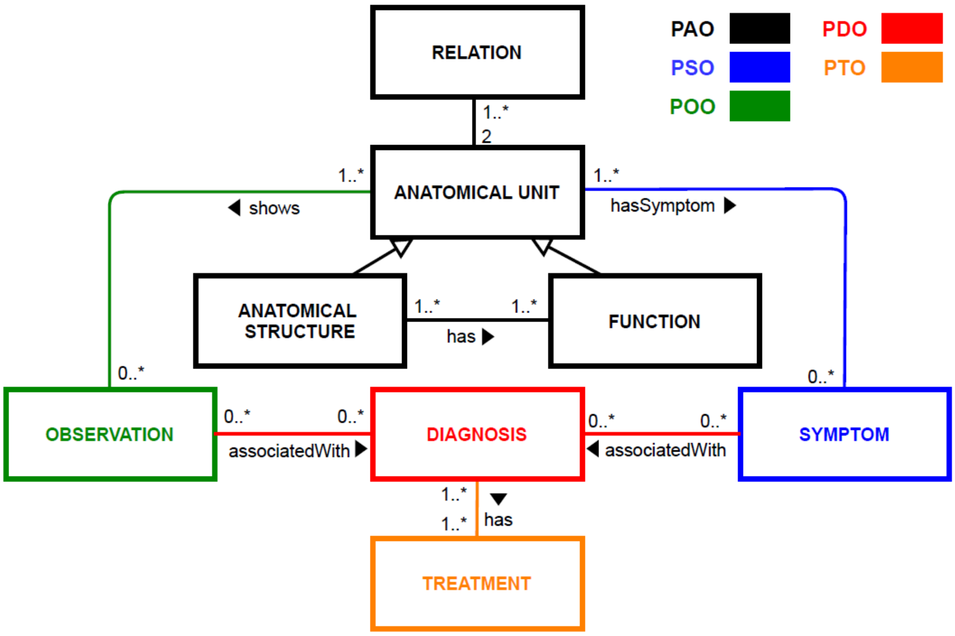


Figure 4.6: Medical guideline ontology (ElAssy et al., 2022)

Creating the MGO

The majority of the MGO is created by running algorithms on the medical guideline. The patient anatomy ontology (PAO) however can be created by utilising the hierarchical SNOMED CT terminology. The anatomical units can be related to each other in a hierarchical way (e.g. ear canal is part-of ears), or to a function with the relation 'has'. The entire process is defined in the eight following steps. Note that while a more detailed explanation existed, for simplicity these are the basic steps to create an MGO (ElAssy et al., 2022).

1. **Target guideline preparation:** The medical guideline is retrieved from the authority's source and prepared for the following activities.
2. **Concept extraction:** Relevant sections for the MGO (symptoms, diagnosis etc) are selected from and all noun phrases are extracted as potential concepts. Not all nouns will be used, but the relevant ones are mapped in the following five activities.
3. **Patient Anatomy Ontology (PAO) construction:** The PAO is constructed by mapping the guideline anatomical nouns to the anatomical concepts of SNOMED CT and converting it to a hierarchy using the aforementioned relations.
4. **Patient Symptoms Ontology (PSO):** The symptoms found in the medical guidelines are mapped to the corresponding anatomical units.
5. **Patient Observations Ontology (POO):** Potential physician's observations from the medical guideline are mapped to the corresponding anatomical units.
6. **Patient Diagnosis Ontology (PDO):** The medical condition corresponding with the medical guideline is mapped to the symptoms and observations.
7. **Patient Treatment Ontology (PTO):** The possible treatments of the medical condition is mapped against the PDO and any relevant anatomical units.
8. **Medical Guideline Ontology Finalization:** All individual subontologies are combined into one MGO and relations are added.

The medical condition of otitis externa (swimmer's ear) is used to exemplify a complete MGO, depicted in figure 4.7. Starting with the patient anatomical ontology (coloured black), the anatomical unit ear is a part of the external auditory canal, as well as the head, which in turn is part of the patient. The Ear also has the function of hearing. Possible symptoms (blue) associated with otitis externa are ear pain, itching, drainage and hearing loss. According to the medical guideline, the physician is able to observe scars, swelling, flaking and redness at the external auditory canal, as well as a ruptured eardrum. These symptoms, observations and anatomical units are related to the disease (red) of otitis externa. Finally, a treatment can be selected based on the physician's observations. A choice can be made in this case to clean the patient's ears, prescribe ear drops, refer to a specialist or a combination of those. Note that the MGO can be seen as a web of triples (set of three entities) in the form of an anatomy unit, symptom, observation, disease or treatment connected to another one with a relation in between.

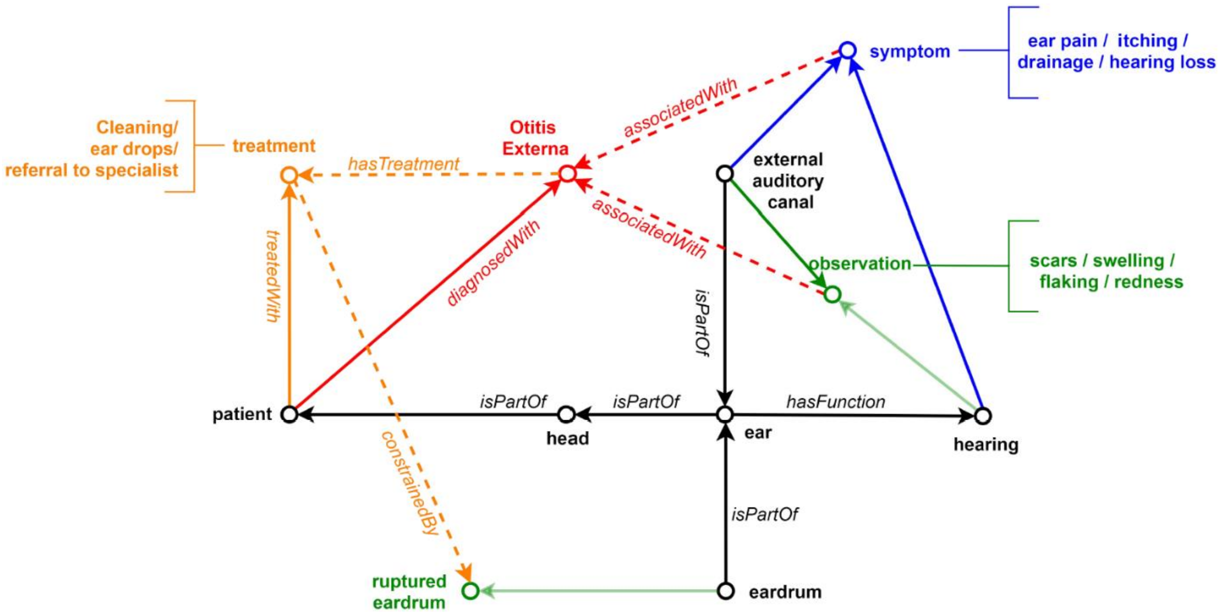


Figure 4.7: medical guideline ontology Otitis Externa (ElAssy et al., 2022)

4.5.2 Conversation interpretation

Assuming the MGO has been created, the consultation conversation must be machine interpreted to generate a report (figure 4.4). This side of the pipeline consists of four steps.

The first step is consultation transcription. The consultations between medical professionals and patients are transcribed using automated speech recognition, which has been discussed in 4.1. The current architecture utilises Google Speech-to-Text to transcribe the conversations.

The next step in the pipeline is creating a consultation knowledge graph (KG) using RDF triples, also called semantic triples. The goal is to interpret the consultation in a machine-readable way so that all relevant information can be retrieved using various NLP techniques. To achieve this, C2R transforms the transcripts to RDF triples in a process called triplication. RDF triples are a set of three entities in the form of {subject, predicate, object} with the purpose of codifying semantic data. Different tools may produce different triples, and further processing may need to be done to get a list of triples accurately reflecting the semantics.

The example of otitis externa will again be used to provide some examples:

- {patient, diagnosedWith, OtitisExterna}
- {ear, hasSymptom, itching}
- {physician, makesObservation, scars}

C2R stores the triples in a knowledge graph platform called Stardog. A knowledge graph in the context of C2R is an instantiated fact base of one consultation (ElAssy, 2022) as opposed to a Medical Guideline Ontology, which is a general schema representing all consultations of that medical condition.
Up until this point, the C2R architecture executed the triplication primarily using FRED. The FRED tool utilises a variety of NLP components to generate RDF triples, which collectively form a knowledge graph. FRED not only utilises linguistic triples, but also enriches the triples with data from ontology languages such as OWL and RDF. As an example the following sentence is used as input: *I am good at running*. This is a simple sentence, but the output as displayed in Figure 4.8 contains a substantial knowledge structure. The graph indicates that running has the theme Thing. Running is also a subclass of the dul ontology Event. Furthermore, running can be seen as a situation involving a person. And both the situation and the person refer to the quality good.

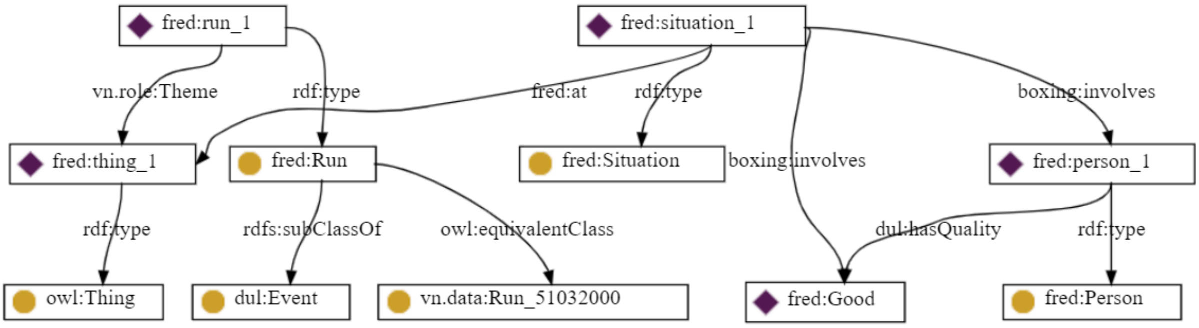


Figure 4.8: Knowledge graph of the sentence: "I am good at running"

FRED is available as an online tool on http://wit.istc.cnr.it/stlab-tools/fred-splash/. The online version of FRED is a particularly useful tool to illustrate the knowledge graph, but other tools such as Stanford OpenIE (Open Information Extraction) can be used to generate lists of triples as well, but taking a more linguistic approach and often leaving out semantics.

Semantic interpretation

After the creation of a consultation knowledge graph and the existence of the Medical Guideline Ontology, the Medical Guideline Ontology is populated with triples of the consultation knowledge graph in a process called semantic interpretation. This process is performed using an algorithm called the triple matching algorithm. By populating the Medical Guideline Ontology, all in the consultation mentioned anatomical units, symptoms, observation, diagnoses and treatments will be populated, while the unmentioned parts of the ontology are not being used. This process ensures that irrelevant information to the medical guideline will be left out, while only the report-essential triples will be used to eventually generate natural sentences and upload them to the EMR. This triples matching algorithm has already been developed, and this algorithm will be adjusted to suit the requirements of geriatric consultations. For now, the functionality of the algorithm is explained through pseudocode in algorithm 4.1 (Maas et al, 2020).

First of all, it is important to understand what a URI is in the context of this code. A URI is a word or word group derived from natural language that serves as an identifier. The MGO consists of a set of entities that may be matched with unmatched triples. These entities are subjects (*S*), predicates (*p*) and objects (*O*). Both subjects (*S*) and objects (*O*) are stored as tuples of URI (*u*) and corresponding natural language nouns (*N*). The five subontologies all have distinct relation types that are used to see which triples are allowed to be matched. For example, the symptoms ontology includes possible relations such as: "subClassOf", "hasQuality", "equivalentClass", "hasSymptom". This collection of possible relations is called *V*.

T is a set of unmatched triples. Unmatched triples consist of subjects (subj), predicates (pred) and objects (obj) as well, however contrary to the set of ontology entities, unmatched triples do not have a URI and consist solely of the natural language nouns found during triplification. *T* is regarded as a graph in which objects and subjects are connected to each other with the predicates. Using a breadth-first search algorithm, the algorithm looks for the first unmatched object. It then checks whether the predicate is allowed by comparing it to *V*. If this is the case, and the object and subject are identical to *N*, then a pair is found and matched by transforming the unmatched triple to the URI (*u)* of *S*, *p* and *O*. These new sets of triples are stored as set *T'*.

```
input  : T = set of input triples (sub, pred, obj)
         S = set of possible subjects, which are tuples of the form (u, N)
           where u is the uri and N a set of natural language nouns
         p = output triple predicate uri
         O = set of possible objects, same tuple format as for S
         V = set of allowed search predicates
output: A new set of triples T′

T′ ← ∅;
for (obu, obN) ∈ O do                          // Search from each object
    visited ← ∅;
    queue ← makeQueue(obN);                    // New queue with initial
      content
    while not empty(queue) do
        ent ← dequeue(queue);
        if ent ∈ visited then                  // Do not visit entity twice
            continue
        add ent to visited;
        found ← false;
        for (sbu, sbN) ∈ S do // Check if entity is subject
            if ent ∈ sbN then
                add triple (sbu, p, obu) to T′; // Triple found
                found ← true;
                break
        if found then                          // Do not explore further
            break
        for (sub, pred, obj) ∈ T do            // Find connected
          entities
            if not pred ∈ V then // Check allowed relations
                continue
            if ent = sub then
                enqueue(queue, obj);
            else if ent = obj then
                enqueue(queue, sub);
```

Algorithm 4.1: triples matching algorithm

5. Geriatric dialogues analysed

To design a solution to the problem discussed in chapter 3, the triples matching algorithm described in section 4.4.3 will be redesigned specifically for the scope of geriatrics. This study will contribute to previous thesis studies conducted by two other Business Informatics students of Utrecht University: Kendall Kemper and Rick Oostveen. Therefore, this chapter will start with an overview of geriatric assessments in the Netherlands, as well as the work of both aforementioned students. In 5.2 we will analyse questions and answers from all transcripts of three specified ADL categories. 5.3 will discuss the questions and answers from three specified IADL categories. Finally in 5.3 we will discuss patterns and conventions in CGA reporting.

5.1 Comprehensive Geriatric Assessment (CGA)

Geriatrics is the medical field that specialises in diseases and disabilities commonly found in the elderly. These diseases are often part of the aging process and can therefore be complex: memory loss, incontinence and impaired mobility to name a few. In clinical geriatrics, assessments are done to evaluate the physical, functional and social problems of older patients. The primary clinical assessment tool in the Netherlands is the Comprehensive Geriatric Assessment (CGA) (Federatie Medisch Specialisten, 2021). The CGA is a multidisciplinary assessment, which means that it involves multiple healthcare specialists. The CGA takes about two hours and consists of the following components:

A. General

1. Medical history
2. Medication

B. Anamnesis

1. **The specific anamnesis (ADL/IADL)** and tractusanamnesis
2. The hetero anamnesis
3. The functionele anamnesis
4. The social anamnesis

C. Physical examination

1. General physical and neurological examination
2. Physiatric examination
3. Functional examiniation

D. Additional research

1. Measurement tools
2. Lab research
3. ECG

In an ideal situation every component of the CGA would be automated. However, this first case study only focusses on *the specific anamnesis*, due to its exact measuring instruments (questionnaires) and relatively simple nature (ordinal scales). The specific anamnesis includes the Algemene Dagelijkse Levensverrichtingen (ADL), which translates to General Daily Life Activities. The ADL measures how dependent or independent patients are when executing day-to-day activities. Furthermore, the specific anamnesis includes the Instrumental Activities of Daily Living (IADL). Similar to the ADL, the IADL measures how independent patients are in certain areas. However, the IADL focusses on areas in and around the household specifically. The combination of ADL and IADL shall be referred to as (I)ADL throughout this thesis.

ADL

The ADL is usually measured using the Barthel index. The Barthel index was developed in 1965 by Barthel and Mahoney to assess people whose impairments interfered with the independent use of their limbs (Mahoney & Barthel, 1965). In clinical geriatrics, it is used to quantify patients' functionality in a range of disorders, while also evaluating if medical or therapeutical interventions are deemed necessary (Federatie Medisch Specialisten, 2021). The list consists of ten functions with activities of daily living that are assessed by the caretaker and then quantitatively rated. Even though there is some variation in how scores are assigned by various geriatrics departments internationally, the following index is most widely used, including by the Radboud UMC Nijmegen. Each function can be rated 0-1, 0-2, or 0-3 depending on the item. The Barthel index used by the geriatrics department in this research is shown in table 5.1.

IADL

The IADL is measured using an eight point scale (Lawton & Brody, 1969). Scoring of the eight functions differs from ADL in the sense that patients can score either a 0 (unable) or a 1 (able) on each function. However, asterisks (*) denote the extent to which a patient can accomplish a task when certain conditions apply. An example for the function *telephone usage*: A score of 1 indicates that the patient can use a phone on own initiative. A score of 1* indicates that the patient only dials some known numbers. The Lawton IADL scale used in this study is shown in table 5.2.

Table 5.1: Barthel index (ADL)

Function/ Organ	**Items**	**Condition**
Bowels	0 = incontinent 1 = occasional accident 2 = continent	• previous week • if the nurse has to give an enema, it is referred to as 'incontinence' • occasional = once a week
Bladder	0 = incontinent or catheter and unable to independently manage. 1 = occasional accident (maximum of once per 24 hours) 2 = continent (during more than 7 days)	• performance last week • a patient with a catheter completely able to manage himself is scored 'continent' • occasional = max once a day
Grooming	0 = dependent 1 = independent: face, hair teeth, shave, washing face	• performance previous 24-48 hours • refers to personal hygiene: brushing teeth: putting teeth in and out, grooming hair • caretaker is allowed to supply necessary items
Toilet use	0 = dependent 1 = requires some aid, but can do some things by her/himself	• patient must be able to go to the toilet, undress, clean, dress and leave • aid = patient can wipe and perform some of the mentioned actions
Feeding	0 = unable 1 = requires aid with cutting, eating, smearing butter etc.	• patient is able to eat normal food • others are allowed to cook and serve • but not made into smaller pieces • aid = food is made into smaller pieces

Transfers	0 = unable 1 = substantial assistance (1 or 2 people physically) 2 = minimal assistance (physically or verbally) 3 = independent	• patient transfers from bed to chair and vice versa • dependent = not able to sit, two people must carry • substantial assistance = 1 strong trained individual or 2 less trained individuals • minimal assistance = 1 individual assists without much effort; or: • supervision for safety
Mobility	0 = unable to relocate 1 = independent with wheelchair, including corners etc. 2 = walks with aid of one person (physically or verbally) 3 = independent, but allowed to use aid of a tool like a walking stick.	• refers to relocating indoors or in the ward; patient is allowed to use aid • patient in wheelchair should be able to take corners and go through doors • aid = an 'untrained' person assists; including giving moral support
Dressing	0 = dependent 1 = requires aid, but able to do half independently 2 = independent zippers etc., can independently put on several garments.	• patient must be able to choose clothing and dress • clothing can be adjusted • half = patient has help with buttons
Stairs	0 = unable 1 = requires aid (physically, verbally, carrying aiding tool) 2 = independent up and down	• the patient Is supposed to be carry any aid to be scored as independent
Bathing	0 = dependent 1 = independent	• usually hardest activity • patient must be able to get out of the bath and wash without supervision • independent when showering = without supervision/without aid

Table 5.2: Lawton's IADL scale

Function	**Items**
Telephone usage	1 = operates the phone on own initiative, dials own number etc. 1* = dials some known phone numbers 1** = answers the phone but does not dial 0 = makes no use of the phone
Shopping	1 = does all groceries independently 0 = does some groceries independently 0* = requires aid when doing groceries 0** = unable to do groceries
Preparing food	1 = able to independently plan and prepare food 0 = able to prepare meals if various ingredients are supplied 0* = able to prepare meals except for diet meals 0** = needs catered and served meals
Housekeeping	1 = able to keep house clean independently, but has help with heavy tasks. 1* = performs light housekeeping tasks. For example: dishes and making beds 1** = performs light housekeeping tasks, but not sufficient 1*** = needs help with all housekeeping activities.

	0 = unable to do housekeeping activities
Doing laundry	1 = does all the laundry 1* = does small laundry activities like washing socks 0 = the entire laundry has to be done by others
Using transportation	1 = travels independently using public transport or own car 1* = arranges taxi transport, but makes no use of other public transport 1** = travels with public transport accompanied by others 0 = limited transport with taxi or cars with help from others 0* = does not travel at all
Handling medication	1 = is responsible for taking the right amount of medication at the right moment 0 = takes responsibility if the right amount of medication is readied in advance 0* = unable to dose own medication
Handling finances	1 = handles financial matters independently (writing cheques, paying bills, go to bank) 1* = handles daily shopping, but requires help with banking and large purchases 0 = unable to manage finances

Data collection at Radboud UMC Nijmegen

Researcher Kemper, K. collaborated with the geriatrics department at the Radboud UMC Nijmegen to record audio of (I)ADL sessions of thirteen patients. Kemper then transcribed those recordings and kept twelve of the transcripts for her study (Kemper, Brinkkemper & Dalpiaz, 2021). A follow-up research by Oostveen, R. resulted in a structured Excel spreadsheet in which each sheet row represents one speaking turn between the doctor and patient (Oostveen, Brinkkemper & Dalpiaz, 2022). Oostveen analysed the transcripts to determine how geriatricians ask questions about each item on the Barthel and Lawton indices, as well as how patients respond to those questions. To appropriately assess the patient, the to-be-designed algorithm should be able to grasp the questions and replies. Therefore, four aspects of the (I)ADL will be examined. Oostveen has already done some of this work, which will be built upon and compared to in this analysis. The four aspects are:

1. Patterns and conventions in questions
2. Patterns in answers
3. Patterns and conventions in reporting
4. Calculation and interpretation

To match the appropriate evaluation to the patient's responses, we must first examine how questions are framed. Which terms appear in questions indicating which item is being evaluated? The amount of questions asked before a patient typically answers will also be mapped out. Following that, we must examine the responses of the patients. What are the keywords and synonyms that are being used? The amount of detail and nuance required for a patient to fully answer a question will be analysed. Following that, it is critical to examine how doctors report their findings. Here we will look at which patient responses will lead to which evaluation. The method of reporting will also be discussed. Finally we have to analyse how the end result is calculated. How does the report assessment lead to a certain score.

5.1.1 Patterns and conventions in questions

During the specific anamnesis, not all (I)ADL topics are discussed explicitly. Depending on the practitioner and the patient, the extent to which the specific anamnesis is discussed varies. In some cases, scores are assumed based on physical appearance, medical records or answers given during other parts of the CGA. The focus of this case study will be completely on (I)ADL, hence only those transcripts will be used, and no other type of material will be used. Table 5.3 and Table 5.4 provide an overview of which (I)ADL categories were explicitly mentioned in the transcripts. Table rows highlighted in green represent the most mentioned categories, whereas red highlighted columns represent the least mentioned categories. For ADL, the majority of the functions were explicitly covered in around half of the transcripts. The most explicitly mentioned function was *Mobility*, while the least mentioned category was *Feeding*. Almost all IADL functions were mentioned in the transcripts. *Handling finances* was the only function covered in all transcripts, while *Laundry, Transportation* and *Medication* were discussed in the least amount of transcripts (9 In total).

Table 5.3: ADL – frequency explicit mentioning of items

Function	Not mentioned	Mentioned
Bowels	6	6
Bladder	6	6
Grooming	5	7
Toilet use	5	7
Feeding	9	3
Transfers	7	5
Mobility	3	9
Dressing	4	8
Stairs	5	7
Bathing	6	6

Table 5.4: IADL – frequency explicit mentioning of items

Function	Not mentioned	Mentioned
Telephone usage	1	11
Shopping	1	11
Preparing food	2	10
Housekeeping	1	11
Doing laundry	3	9
Using transportation	3	9
Handling medication	3	9
Handling finances	0	12

Legend Most frequently mentioned
Least frequently mentioned

Restriction to most extensively discussed categories

In this case study, we will focus on automating three ADL and three IADL categories, by selecting three categories of each index to analyze in greater detail. The six picked functions are:

ADL
- Mobility
- Dressing
- Stairs

IADL
- Shopping
- Housekeeping
- Finances

These functions were chosen due to them being more frequently mentioned in the transcripts and generally having more textual data. The next step is to determine how many questions a doctor must ask for each item before making an accurate assessment. Oostveen identified two types of questions while analysing the transcripts: questions and follow-up questions. Questions are the initial doctor's questions to begin evaluation of an item. Follow-up questions are used to specify questions or elicit clarification of the initial patient's answer. Table 5.5 and Table 5.6 show the frequency of questions asked during each item session for all transcripts combined. For most items there are equally as many initial questions as follow-up questions, meaning that each item on average requires one initial question and one follow-up question to make an assessment. The one exception is Dressing, where almost no follow-up questions were required. Note that there may be variance between individual transcripts, meaning that during some consultation one question might have been asked, whereas others may have had three questions.

Table 5.5: Question frequency ADL

Item	Initial questions	Follow-up questions	Total	Misunderstanding
Mobility	9	10	19	0
Dressing	8	2	10	0
Stairs	7	8	14	0

Table 5.6: Question frequency IADL

Item	Initial questions	Follow-up questions	Total	Misunderstanding
Shopping	10	10	20	0
Housekeeping	11	11	22	0
Finance	12	15	27	0

Oostveen considered the initial questions to be the most significant as those identify the item and elicit explicit answers. Therefore, those questions are the most important ones to be processed by the algorithm. Oostveen furthermore identified four types of initial questions:

1. *Literal (L):* Literal question about an item as described in the index.
2. *Paraphrased (P):* not literal, with synonyms.
3. *Suggestive (S):* Includes an expected answer.
4. *Confirmation (C):* A subset of suggestive questions which are a confirmation of a previous answer.

For each initial question, the question type was identified and the findings are summarized in Table 5.7 and Table 5.8 for ADL and IADL respectively. Each letter represents the question type: Literal (L), Paraphrased (P), Suggestive (S) and Confirmation (C). The first thing to notice is the general lack of literal questions. More often than not, questions are paraphrased or asked in a suggestive manner. Due to the high amount of paraphrased questions, the algorithm should recognise and handle synonyms. The algorithm should for example not only find objects or subjects named finance, but also words like bills, iDeal, payments and more complex sentence structures. These synonyms, and examples of the question types, will be explored further once each individual item is analysed. The high amount of suggestive answers can indicate that answers are often already given or derived from previous questions or from other sources.

Table 5.7: Frequency question type ADL

Item	L	P	S	C	T
Mobility	1	4	3	0	8
Dressing	3	1	4	0	8
Stairs	1	4	1	0	6

Table 5.8: Frequency question type IADL

Item	L	P	S	C	T
Shopping	3	4	2	1	10
Housekeeping	1	6	4	0	11
Finance	2	6	4	0	12

5.1.2 Patterns in answers

During the (I)ADL assessment, answers are given by both patients and caregivers. Oostveen identified three types of answers:

1. *Explicit (E):* yes or no answer.
2. *Explicit with explanation (e):* yes or no, followed by an explanation.
3. *Implicit (I):* a longer more complex answer without a literal yes or no, but out of which an answer can be deduced.

For each answer to the initial questions, the answer types were identified. The findings are summarised in Table 5.9 and Table 5.10. The findings show that during the discussion of ADL items patients generally give explicit answers or explicit answers followed by an explanation. That is, a combination of the doctor's question and a confirmation or denial answer should generally provide the data necessary for an assessment. The item stairs differs from other ADL items in the sense that there are some implicit answers. Again, examples and a more in depth analysis of each item is presented later. During IADL assessments there are in general less explicit answers. Implicit answers will be more complex to be handled by the algorithm due to synonyms and different grammatical structures. In the following parts we take a closer look at the six chosen item

Table 5.9: Frequency answer type ADL

Item	E	e	I	T
Mobility	3	2	0	5
Dressing	5	3	0	8
Stairs	4	1	3	8

Table 5.10: Frequency answer type IADL

Item	E	e	I	T
Shopping	1	5	9	15
Housekeeping	5	3	3	11
Finance	3	6	6	15

5.2 Patterns in ADL: questions and answers

The transcripts of the three chosen ADL categories are discussed in this chapter. We will look at both question and answer patterns in the transcripts. Furthermore, key indicators for patient's assessments and categories are identified and discussed. Finally, a comparison will be made between the geriatric assessment scores and expected scores based on analysing patient answers.

5.2.1: Patterns in Mobility

For mobility, most questions were paraphrased. All questions were answered in an explicit manner. Table 5.11 shows all transcripts for this item. Yellow words are unique to the item being discussed. These words are verbs (actions) or nouns (objects) that are usually used during these actions. The primary word used to indicate an action for the item mobility was 'lopen'. 'Vallen' was used as well by doctors to ask whether the patient can walk without falling. Other words used in this category were 'rollator' and 'stok' which are different types of walking aid. Blue words are dependency indicators, words indicating whether a patient requires aid or is able to perform the item independently. 'Hulpmiddel' (aid) is the most common dependency indicator used by doctors in this category. The patients respond with either an explicit answer or 'zonder' (without). For this category, we saw a combination of simple questions combined with explicit answers. Usually the first question inquired about whether the patient could work or not, whether the second question inquired about any walking aid.

Table 5.11: ADL Mobility complete transcripts

No.	Actor	TEXT	Type	Question type	Answer Type
T1	D	En het lopen? Loopt u zonder hulpmiddel of heeft u een stok of een rollator?	Question	Suggestive	
T1	P	Nee, zonder.	Answer		Explicit
T1	D	Helemaal zonder. En valt u weleens?	Follow-up question		
T1	P	Nee.	Follow-up answer		
T1	D	En hoe gaat het lopen?	Follow-up question		
T1	P	Goed.	Follow-up answer		
T2	D	Ah, fijn. Oke, kan u lopen zonder hulpmiddel?	Question	Literally	
T2	P	Ja.	Follow-up answer		
T2	D	Of heeft u een stok of een rollator? Nee he?	Question	Suggestive	
T2	P	Nee.	Follow-up answer		
T2	M	Niet meer. [onhoorbare tekst].	Follow-up answer		
T2	P	Ja, nou, na mijn heupoperatie.	Follow-up answer		
T3	D	Ja. En als u moet lopen, ik zie een rollator, is die van u?	Question	Paraphrased	
T3	M	Nee, dat is de mijne.	Answer		Explicit w explanation
T3	D	Oh, ja ik wou zeggen dat kan ook nog. Heeft u zelf een hulpmiddel, een rollator of een stok?	Follow-up question		
T3	P	Ik heb ze wel, maar niet bij me.	Follow-up answer		
T3	D	Nee. Gebruikt u het überhaupt?	Follow-up question		
T3	P	Ja.	Follow-up answer		
T3	D	Thuis wel.	Follow-up question		
T3	P	Ja.	Follow-up answer		
T3	D	Oke, ook binnen het huis gebruikt u het?	Follow-up question		
T3	P	Ja.	Follow-up answer		
T3	D	Oke. En dan hoe ver kan u lopen ongeveer? Kan u nog lopen buiten?	Follow-up question		
T3	P	Zeker.	Follow-up answer		
T3	D	En hoe gaat, want dan loopt u met een rollator, kan u, hoe ver kan u komen?	Follow-up question		
T3	M	Nee niet met de rollator, maar met de stok.	Follow-up answer		
T5	D	Oke, en, en heeft meneer ook een rollator of iets?	Question	Paraphrased	
T5	M	Ja, hij heeft wel rollator gekregen, maar hij, maar hij komt wel van boven, van slaapkamer naar beneden.	Answer		Explicit w explanation

T6	D	Nee precies, en, ja, het lopen met de stok hebben we ook besproken. De rollator is eigenlijk geen optie he met die schouder?	Question	Suggestive	
T6	P	Nee.	Answer		Explicit
T8	D	Oke. En verder heeft u een rollator of een stok?	Question	Paraphrased	
T8	P	Nee.	Answer		Explicit
T8	D	Of eigenlijk helemaal geen hulpmiddel?	Follow-up question		
T8	P	Nee.	Follow-up answer		
T8	D	Nee, en wat kunt u zo al lopend trouwens? Voordat u zo helemaal uitgeput bent. Kunt u vijfhonderd meter lopen of?	Follow-up question		
T8	P	Jawel.	Follow-up answer		

Table 5.12 shows the scores for each transcript for this item. GA (geriatric assessment) is the score assigned to the patient by the doctor. The TS (transcript score) is the expected value of the assessment based only on the transcripts. As mentioned before, doctors have more data available than us, as well as visual observations. It is therefore possible that mismatches occur between the used language in the transcripts and the final score. The conversation interpretation algorithm will not be able to correctly assess these instances. These differences in assessment will therefore be summarized and analysed for each category in these tables. As table 5.12 shows, the transcripts and final doctor assessments were consistent and no differences were found. Each patient scored three points, meaning that the patients are able to independently relocate, possibly with the aid of a rollator or walking stick.

Table 5.12: Reported assessment and transcript scores: Mobility

Transcript	GA	TS	Dif.
Transcript 1	3	3	
Transcript 2	3	3	
Transcript 3	3	3	
Transcript 5	3	3	
Transcript 6	3	3	
Transcript 8	3	3	

5.2.2: Patterns in Dressing

For dressing, most questions were asked in a suggestive manner. This suggests that the doctor already has pre-existing knowledge, either through earlier conversations or medical records. The patients all gave explicit answers (some with explanation) to those questions. The most common terms are 'aan- en uitkleden' or a conjugation of those words. Noticeable is that the verbs 'wassen' and 'douchen' are also mentioned, which belong to the separate item bathing, meaning that doctors usually inquire about these items in one batch of questions. 'Helpen' (helping) and 'zelf' (self) are dependency indicators used for most items. 'Thuiszorg' (home care) are specific to a few categories. Patients' answers are relatively simple using words as yes, no or zelf.

Table 5.13: ADL Dressing complete transcripts

No.	Actor	TEXT	Type	Question type	Answer Type
T3	D	Ja, oke. Ja. En het aan- en uitkleden helpen ze daar ook bij?	Question	Literally	
T3	P	Nee.	Answer		Explicit
T3	P	Ja dat wil zeggen met wassen en uitkleden. Dat natuurlijk wel.	Follow-up answer		
T3	P	Dat soort hulp.	Follow-up answer		
T3	D	Ja, dus dan helpt wel iemand u met het aan- en uitkleden?	Follow-up question		
T3	P	Ja.	Follow-up answer		
T4	D	Ja, en dat lukt u? Ja. En bijvoorbeeld als u ergens, als u aan en uit moet kleden of douchen bijvoorbeeld?	Question	Literally	
T4	P	Ja.	Answer		Explicit
T4	D	Lukt u dat zelf? Of komt de thuiszorg?	Follow-up question		
T4	P	Ik kan dat zelf. Want ik doe het. Ik word, twee keer in de week, word ik gedoucht door de thuiszorg.	Follow-up answer		
T4	P	En dan tussendoor doe ik het zelf, maar dat duurt dan gewoon iets langer als normaal zeg maar he. Ik moet het van mezelf ook heel rustig doen, want ik neem overal de tijd voor, want eigenlijk, ja, kan dat.	Follow-up answer		
T5	D	Hoger geworden, ja, ja. En als hij gaat wassen, aankleden, douchen, heeft hij dan hulp nodig?	Question	Literally	
T5	M	Nee, dat niet.	Answer		Explicit w explanation
T5	M	Dat is wel met moeite.	Follow-up answer		
T5	M	Dat is wel, alles is, zeg maar met moeite, doet die wel.	Follow-up answer		
T5	D	En dus ook aankleden, doet hij dus zelfstandig ook?	Question	Suggestive	
T5	M	Ja, zelfstandig maar met moeite, zeg maar.	Answer		Explicit w explanation
T6	D	En aan- en uitkleden heeft u wel hulp bij he, van de thuiszorg?	Question	Suggestive	
T6	P	Ja, ja, ja.	Answer		Explicit
T7	D	Oh, ja, dan kun je ook naar het toilet, ja. En douchen lukt u ook en aankleden, uitkleden?	Question	Paraphrased	
T7	P	Ja, is niks. Er is niks in het rijtje wat ik niet kan.	Answer		Explicit w explanation
T8	D	Ja, ja, precies. En aan- en uitkleden doet u dus ook zelfstandig he?	Question	Suggestive	
T8	P	Ja.	Answer		Explicit
T9	D	Maar bijvoorbeeld voor uzelf zorgen, dus het wassen, aankleden, dat doet u allemaal zelf?	Question	Suggestive	
T9	P	Ja.	Answer		Explicit
T11	D	Zelf, ja. En ook het wassen, aankleden, doet u allemaal zelf?	Question		
T11	P	Ja, ja.	Answer		

As illustrated in table 5.14, six out of eight patients are able to independently dress, whereas two patients require home help to aid them. The key dependency indicators of 'zelfstandig' and 'thuiszorg' in combination with a explicit answer lead to these assessments. Suggestive questions are often combined with explicit answers, meaning that these question types often result in a small quantity of text. The doctor's consistent use of these dependency indicators along with the straightforward answers, makes triple generation relatively simple.

Table 5.14: Reported assessment and transcript scores: Dressing

No.	GS	TS	Dif.
Transcript 3	0	0	
Transcript 4	2	2	
Transcript 5	2	2	
Transcript 6	0	0	
Transcript 7	2	2	
Transcript 8	2	2	
Transcript 9	2	2	
Transcript 11	2	2	

5.2.3: Patterns in Stairs

Doctors ask about the patients' ability to walk stairs in a paraphrased manner (Table 5.15). Patients primarily respond with an explicit answer, some with explanation. In comparison to the prior two items, however, there were more implicit answers given. The item-specific terms are 'traplopen', 'trap', 'trappenhuis' and 'lift', which translate to stairs and elevators. Again, the common dependency indicators of 'help' and 'zelf(standig)' were used. Whereas patients in the previous two Barthel categories frequently gave single-word responses, this category sees more implicit responses using one or more sentences. For example, one patient responded with *"Wij hebben geen trappen meer"* (We don't have stairs anymore). Curiously, this patient still received an 'independent' rating on this category, indicating that the doctor had alternative ways to assess the patients capacity to walk stairs. Another example of an implicit follow-up answer is *"Ik ga nooit met de lift eigenlijk"* (I never take the elevator), implying that the patient in fact makes use of stairs and is therefor able to walk stairs.

Table 5.15: ADL Stairs complete transcripts

No.	Actor	TEXT	Type	Question type	Answer Type
T1	D	Kunt u trap lopen?	Question	Literally	
T1	P	Ja.	Answer		Explicit
T1	D	Hebben jullie een trap in het huis?	Follow-up question		
T1	M	Ja, het trappenhuis.	Follow-up answer		
T1	P	Ja. Trappenhuis. Ik ga nooit met de lift eigenlijk.	Follow-up answer		
T1	D	Nee dus dat u ook nog en dat lukt ook nog?	Follow-up question		
T1	P	Ja hoor.	Follow-up answer		
T2	D	Ja eten lukt u ook en een trap lopen zou dat lukken?	Question	Paraphrased	
T2	P	Ja.	Answer		Explicit

T2	D	Hebben jullie, wat voor woning hebben jullie?	Follow-up question		
T2	M	De eerste etage in een appartement. We lopen bijna altijd met de trap. [onhoorbare tekst].	Follow-up answer		
T2	P	Ik neem alleen maar, ik nam alleen maar de lift als ik thuis kwam met de boodschappen.	Follow-up answer		
T2	M	Maar die worden nu bezorgd.	Follow-up answer		
T2	P	Maar die worden nu bezorgd.	Follow-up answer		
T4	D	En hebben jullie een trap bijvoorbeeld in het huis?	Question	Paraphrased	
T4	P	Ja wij moeten altijd trap, loop ik ook iedere dag een paar keer. Ik ga 's avonds ook een paar keer op en neer, al hoef ik niks te doen.	Answer		Explicit w explanation
T4	P	Maar ik loop iedere avond wel een paar keer de trap op en neer.	Follow-up answer		
T5	M	Met de trap, wel met moeite, wel naar beneden.	Answer		Implicit
T5	M	Maar, dat is zelfstandig.	Follow-up answer		
T5	D	Was dat voor corona ook zo?	Follow-up question		
T5	M	Ja.	Follow-up answer		
T5	D	Was het toen ook met moeite?	Follow-up question		
T5	D	Voor corona en voor?	Follow-up question		
T5	M	Niet echt, maar wel ja, voor sowieso ouderen.	Follow-up answer		
T5	M	Maar doet ie wel alles, dat zelfstandig, maar.	Follow-up answer		
T5	D	Ja, ja. En de trap af kost ook moeite. Trap op?	Question	Suggestive	
T5	M	Ja.	Answer		Explicit
T6	D	Trappen lopen kunt u dat?	Question	Paraphrased	
T6	P	Heel perfect.	Answer		Implicit
T6	P	Ja, ik loop echt goed trap he? Ja je verwacht het niet, maar het is wel zo.	Follow-up answer		
T7	D	Trap lopen ook?	Question	Following	
T7	M	wij hebben geen trappen meer.	Answer		Implicit
T8	D	Daar heeft u geen hulp bij. Heeft u trappen thuis?	Question	Paraphrased	
T8	P	Ja.	Answer		Explicit
T8	D	Ja en trappen op, af?	Follow-up question		
T8	P	Ik heb een heel huis, ja.	Follow-up answer		
T8	D	Ja, ja, en dat gaat probleemloos? Die trappen op en af of heeft u?	Follow-up question		
T8	P	Nog wel.	Follow-up answer		

As was the case with the other two ADL items, no inconsistencies were found between the transcripts and the doctor's assessments (Table 5.16). All patients scored a 2, meaning that they are able to independently walk stairs. In some cases, these assessments are straightforward with literal questions and explicit answers. Transcript 1 and 2 are two examples of these. However, in transcripts 4 and 8, the doctor asks the patients indirectly if they have stairs at home. Patient 4 says they have to take the stairs all the time, and patient 8 says they have a big house.

Table 5.16: Reported assessment and transcript scores: Stairs

No.	GS	TS	Dif.
Transcript 1	2	2	
Transcript 2	2	2	
Transcript 4	2	2	
Transcript 5	2	2	
Transcript 6	2	2	
Transcript 7	2	2	
Transcript 8	2	2	

5.3 Patterns in IADL: questions and answers

The transcripts of the three chosen IADL categories are discussed in this chapter. Similar to 5.2, patterns in question and answers, key indicators and the doctor's assessments will be discussed.

5.3.1: Patterns in Shopping

The analysed transcripts of the category shopping are found in Table 5.17. Compared to the ADL items, IADL items are relatively more complex. A higher number of synonyms and item specific terms are used. Generally, more information is required to make an assessment. Questions were asked most often in a paraphrased manner, although the other question types occurred as well. Most given answers were explicit, contrasting those of the ADL items. The item-specific terms are 'boodschappen', 'briefje', 'product', 'voorraden', 'kopen', 'bestelling' and 'winkel'. These terms are all related to shops, products and orders. While for ADL items we usually see the same dependency indicators of 'zelf' or 'help', for shopping additional dependencies in the form of friends and family are found. For example, one patient's wife and another patient's daughter went shopping for them. In addition to that, one patient is dependent on ordering products online.

Table 5.17: IADL Shopping complete transcripts

No.	Actor	TEXT	Type	Question type	Answer Type
T1	D	En ook de telefoon aannemen. Boodschappen doen?	Follow-up question		
T1	P	Schrijf ik op een briefje.	Follow-up answer		
T1	D	Ja, en dan?	Follow-up question		
T1	P	Dan gaat het nog mis.	Follow-up answer		
T1	D	Gaat het dan nog mis?	Follow-up question		
T1	M	Dan is hij dingen vergeten.	Follow-up answer		
T1	D	Oh ja. En dan thuis dan komt u erachter of?	Follow-up question		
T1	M	Eén boodschap gaat, maar twee gaat dan niet meer.	Follow-up answer		
T1	D	Eén boodschap als in één product?	Follow-up question		
T1	M	Eén product.	Follow-up answer		
T1	D	Ja, oke, ja. Maar u doet het dus wel?	Follow-up question		
T1	P	Ja.	Follow-up answer		
T1	D	Nee. Het is niet dat u hele voorraden van een bepaald product heeft staan en dan nog meer daarvan koopt?	Follow-up question		

T1	P	Nee, nee.	Follow-up answer		
T1	M	Alleen laatst met die broodjes. Dat ging mis bij de bakker. Moest hij de bestelling ophalen. Dat is een bruin brood en twee zakken broodjes, maar ik had iets waardoor we nog meer broodjes nodig hadden dus [onhoorbare tekst] extra broodjes. Dan had hij dus, de eerste keer was je dat vergeten en de tweede keer toen je naar de bakker moest, de week daarop, toen bracht hij en die zak extra plus nog drie zakken extra. Sindsdien heb ik zesendertig broodjes.	Follow-up answer		
T2	D	En boodschappen, die worden dus bezorgd en wie bestelt de boodschappen?	Question	Suggestive	
T2	M	Hij bestelt alles op de computer.	Answer		Implicit
T2	P	Ja, en ik pak zelf altijd de boodschappen uit.	Follow-up answer		
T2	D	Gaat dat goed of hebben jullie twintig dingen van hetzelfde en is alles mis?	Follow-up question		
T2	P	Nee, nee, nee.	Follow-up answer		
T2	M	Nee, nee, alleen af en toe kleine dingen vergeten.	Follow-up answer		
T2	P	Ja, dan wil jij, dan heb je een keer gezegd van wil je dat doen.	Follow-up answer		
T3	D	En boodschappen doen, wie doet dat van jullie? De boodschappen.	Question	Literally	
T3	P	De boodschappen, dat gebeurt door of mijn vrouw of door iemand anders. Hulp genoeg.	Answer		Explicit w explanation
T3	D	Hulp genoeg, ja. Dus u doet de boodschappen vooral?	Follow-up question		
T3	M	Af en toe doe ik het één of ander, maar, of degene(n) die komen helpen doen de boodschappen.	Follow-up answer		
T4	D	En hoe doet u het bijvoorbeeld met de boodschappen?	Question	Paraphrased	
T4	P	Nou, die doen we samen.	Answer		Implicit
T5	D	En de boodschappen doet hij niet. Oh, ja.	Question	Confirmation	
T5	M	Ik heb dat zelf niet gezien, omdat.	Answer		Implicit
T6	D	Helemaal goed. Ja, en de boodschappen, doet u dat?	Question	Literally	
T6	P	Dat doet deze dame voor mij.	Answer		Explicit w explanation
T6	D	Ja, dus eigenlijk doet u geen, ook kleine boodschapjes, niet he?	Follow-up question		
T6	P	Ik doe helemaal niks nee, nou ja dat klinkt.	Follow-up answer		
T6	M	Nee, maar dat gaat ook helemaal niet meer.	Follow-up answer		
T6	P	Nee, nee.	Follow-up answer		
T7	D	Het, de boodschappen doet u die zelf?	Question	Literally	
T7	M	Hij doet de boodschappen.	Answer		Implicit
T7	P	Ik doe alle boodschappen.	Answer		Implicit
T8	D	Ja, dat komt vanzelf wel weer terug, ja. En boodschapjes doen? Doet u dat zelf of doet iemand dat voor u?	Question	Paraphrased	
T8	P	Doe ik zelf.	Answer		Implicit
T8	P	Ja. Ik ga zelf ook boodschappen doen.	Answer		Explicit w explanation
T8	P	Ik heb een mooie boodschappenwagentje gekregen van hun dat ik vooruit kan duwen.	Answer		Implicit
T8	D	Oh, hartstikke goed.	Follow-up answer		

T9	D	Ja, en de boodschappen doet u samen?	Question	Suggestive	
T9	M	Dat doen we samen, ja.	Answer		Explicit w explanation
T9	P	Ja.	Follow-up answer		
T9	P	Ik denk dat jij de meeste boodschappen doet? De dichtstbijzijnde, zeg maar.	Answer		Implicit
T11	D	Ja, ja, ja. En boodschappen doet u, u zegt nou u gaat wel in de buurt gaat u wel zelf op pad. En kleine boodschapjes doen, doet u dat nog zelf?	Question	Paraphrased	
T11	P	Ja, aan het begin van corona. Toen mijn dochter zegt, mama jij blijft thuis ik doe de boodschappen en onhoorbare tekst] ook af en toe.	Answer		Explicit w explanation
T12	D	Uhu. Oke. Even kijken want u, u de boodschappen, dat doet u allemaal zelf. Ook de grote boodschappen of krijgt u daar wel eens hulp van?	Question	Paraphrased	
T12	P	Ja, wat zijn grote boodschappen? Ja, dat kun je verdelen over de hele week, dan zijn het kleintjes.	Answer		Implicit
T12	P	Maar ik ga niet iedere dag naar de winkel hoor, dat zeker niet.	Follow-up answer		

Table 5.18 indicates the doctors assessments for IADL function shopping. Each possible item has been scored by one or more patients. The patients of transcript 1 and 9 are independently able to do small groceries. The key sentence in transcript 1 is: Een product gaat, maar twee niet meer. The patient indicates that he or she is able to buy one product, but starts to forget when buying more. Patients 2, 7, 8, 11 and 12 all score a 1 (independent). The main dependency indicator is 'zelf'. Patients 3, 5 and 6 are entirely dependent on someone else for groceries. Patient 3 states he receives help from multiple sources. Patient 6 indicates a carer does groceries for him.

There are two inconsistencies between expected values, and doctor's assessment. The doctor valuated that the patient corresponding with transcript three was entirely unable to go shopping (0**), while the transcript suggested they may be able to with aid (0*). Due to the translated sentence *"Every now and then I do something"* the assumption was made the patient is still able to do some groceries, however, the doctor estimated that this was not enough to give a 0*. For transcript 9, it was presumed the patient needs assistance given that the patient mentioned doing groceries with the carer and that the carer does the majority of the groceries. However, the doctor decided that the patient is capable of doing minor grocery shopping independently.

Table 5.18: Reported assessment and transcript scores: Shopping

No.	GS	TS	Dif.
Transcript 1	0	0	
Transcript 2	1	1	
Transcript 3	0**	0*	*
Transcript 4	0*	0*	
Transcript 5	0**	0**	
Transcript 6	0**	0**	
Transcript 7	1	1	
Transcript 8	1	1	
Transcript 9	0	0*	*
Transcript 11	1	1	
Transcript 12	1	1	

5.3.2: Patterns in Housekeeping

Questions about housekeeping were asked with a paraphrased approach. Patients gave primarily explicit answers, although implicit more substantial answers were given as well. Item-specific terms either consisted of the item name 'huishouden', but also specific housekeeping tasks like 'stofzuigen' and 'afwassen'. The primary dependency indicator for housekeeping is home help ('huishoudelijke hulp') For this category's assessment it is important to figure out what tasks the patient is able to do independently and for which tasks the patient needs help. Based on the difficulty of these tasks the patient will receive a different rating. In certain cases, the patient will immediately respond that they have home help for everything, or that they are perfectly capable of doing housekeeping. In other cases however, the doctor needs to ask follow-up questions to acquire all the details. An example can be found in transcript 3. The patient reveals in the first answer that home help is required for housekeeping, but during the next question it is revealed that certain tasks, like loading the dishwasher is still possible. Instead of receiving the score 0, this patient receives a 1*** due to this answer.

Table 5.19: IADL Housekeeping complete transcripts

No.	Actor	TEXT	Type	Question type	Answer Type
T1	D	Het huishouden deed u wel he?	Question	Suggestive	
T1	P	Ja.	Answer		Explicit
T1	D	En hebben jullie dan ook huishoudelijke hulp daarnaast of doet u eigenlijk alles zelf of?	Follow-up question		
T1	P	Ja.	Follow-up answer		
T2	D	Het huishouden, kan u daar iets in doen?	Question	Paraphrased	
T2	P	Ja kan wel, maar het gebeurt nooit.	Answer		Explicit w explanation
T2	D	Nee?	Follow-up question		
T2	P	Ik kan wel stof zuigen.	Follow-up answer		
T2	D	Doet u het weleens of niet?	Follow-up question		
T2	P	Ja, nou als de hulp een langere tijd, een langere tijd niet komt, dan zegt [naam mantelzorger] nou.	Follow-up answer		
T2	D	En afwassen? Afwassen of het bed even.	Follow-up question		
T2	M	De vaatwasser.	Follow-up answer		
T2	P	Ja, de vaatwasser ook wel. Ja, maar.	Follow-up answer		
T2	M	Leeg ruimen.	Follow-up answer		
T3	D	En het huishouden? Dat zei u al dat jullie hulp hebben, huishoudelijke hulp.	Question	Suggestive	
T3	P	ja.	Answer		Explicit
T3	D	En doet u bijvoorbeeld nog wel zoiets als de afwas of ik weet niet of jullie een vaatwasser hebben leegruimen of ook niet?	Follow-up question		
T3	M	Nou, niet leegruimen, maar je ruimt hem wel altijd in he?	Follow-up answer		
T3	M	En de keuken opruimen, dat is een beetje een [onhoorbare tekst].	Follow-up answer		
T3	P	Ja, ja.	Follow-up answer		
T4	P	En een keer in de week hebben we hulp.	Answer		Implicit
T4	D	Oke, huishoudelijke hulp?	Question	Literally	

T4	P	Ja.	Answer		Explicit
T4	D	Ja, en de kleinere huishoudelijke dingen, dus afwassen, doekje over de tafel?	Follow-up question		
T4	P	Ja, dat doet hij.	Follow-up answer		
T5	D	Ja. En qua huishouden, doet hij niks in huis?	Question	Suggestive	
T5	M	Nee.	Answer		Explicit
T5	D	Heeft hij ook nooit gedaan?	Follow-up question		
T5	M	Nee.	Follow-up answer		
T6	D	Daar hadden we het ook over gehad en huishouden?	Question	Paraphrased	
T6	P	Nee, daar komt ook deze dame voor.	Answer		Explicit w explanation
T7	D	Ja. Ook het huishouden?	Question	Paraphrased	
T7	P	Nou, dat delen we met een bepaalde.	Answer		Implicit
T7	D	Daar hebben jullie een verdeling? Dat is altijd al zo geweest? Ja. Het huishouden verdelen jullie, he?	Question	Suggestive	
T8	D	Ja, oke, dus dat doet u zelfstandig dat huishouden of krijgt u daar ook hulp bij?	Question	Paraphrased	
T8	P	Ik heb drie uurtjes, drie uurtjes heb ik hulp vrijdags.	Answer		Implicit
T8	D	Oh, dus elke week hulp?	Follow-up question		
T8	P	Ja.	Follow-up answer		
T9	D	Ja. En in het, in het huis. Wat doet u allemaal in het huis? Wat zijn dingetjes die u doet?	Question	Paraphrased	
T9	P	In het huis. Ja, daar ben ik mee bezig gewoon met opruimen.	Answer		Explicit w explanation
T9	P	Ja, dat kan ik wel zeggen.	Follow-up answer		
T9	M	Ja, ik ben de huisman. Ik kook en ik, ik poets de vloer en dat soort dingen.	Follow-up answer		
T11	D	Ja, ja. En heeft u hulp thuis? Komt daar iemand voor het huishouden om u te helpen?	Question	Paraphrased	
T11	P	Nee.	Answer		Explicit
T11	M	Ze doet alles zelf.	Follow-up answer		
T11	P	Ik doe alles zelf.	Follow-up answer		
T12	D	Nee. Ja. En, en klusjes in huis. Als er iets moet gebeuren, doet u dat zelf of krijgt u daar hulp bij?	Follow-up question		
T12	P	Nou, dan doe ik dat zelf ja.	Follow-up answer		
T12	D	Doet u allemaal zelf. Ja.	Follow-up question		
T12	P	Ja, er zijn weinig echte, behalve schoonmaken, zijn er geen directe klussen. Ja, ramen zemen of zoiets, maar dat, zou dat zou kunnen.	Follow-up answer		
T12	D	Ja, ja en ook geen ondersteuning van een, een schoonmaakster of een, een huishoudhulp?	Follow-up question		
T12	P	Nee, eigenlijk niet.	Follow-up answer		

Table 5.20 shows the geriatric scores for the item housekeeping. There are five possible scores in this category, and there are many subtle differences between these scores. Housekeeping seems to be the hardest item to interpret by the algorithm for that reason. Both the most independent (1) and dependent (0) score are relatively simple to identify. Fully dependent patients indicate to do no housekeeping tasks or receive help, while the dependency indicator 'zelf' is present in the dialogue of most independent patients. The values in between (1*, 1** and 1***) require an explanation of 'light housekeeping tasks', for which there is no standard. In the transcripts, patients or doctors give examples of tasks which might be considered light, for example, such as doing dishes or cleaning the table after dinner. In an ideal world, the conversation interpretation algorithm could distinguish these specific tasks and make an assessment based on these tasks.

One inconsistency was found between doctor's assessments and the transcripts. Transcript 8 indicates that the patient requires assistance with housekeeping activities, as evidenced by the patient receiving three hours of assistance per week. The doctor however determined that the patient is still able to do light work. The definition of performing light tasks is subjective, so inconsistencies between people are bound to happen.

Table 5.20: Reported assessment and transcript scores: Housekeeping

No.	GS	TS	Dif.
Transcript 1	1	1	
Transcript 2	1*	1*	
Transcript 3	1***	1***	
Transcript 4	1**	1**	
Transcript 5	0	0	
Transcript 6	0	0	
Transcript 7	1	1	
Transcript 8	1*	1***	**
Transcript 9	1*	1*	
Transcript 11	1	1	
Transcript 12	1	1	

5.3.2: Patterns in Finances

For finances, most questions were asked in a paraphrased manner. An example of this is as follows. Instead of asking: Are you able to do your own finances? A doctor would ask something like: If you go shopping, how do you pay? Noticeable is the large body of text compared to the other items. This is partly due to the fact that finances are not a single action. Finances consist of paying bills, banking matters, paying in shops, using a credit card. Answers were divided between implicit and explicit answers. Item-specific terms consisted of shops, payment methods and tools and other financial terms: 'winkel', 'betalen', 'contant', 'pinpas', iDeal' etc. The main dependency indicators were (redacted) names of carer, family and friends.

Table 5.21: IADL Finance complete transcripts

No.	Actor	TEXT	Type	Question type	Answer Type
T1	D	Als u in de winkel moet betalen, hoe doet u dat? Met een pin of gewoon contant?	Question	Paraphrased	
T1	P	Gewoon contant.	Answer		Implicit
T1	D	En pint u dan zelf? Van te voren of hoe komt u aan contant geld?	Follow-up question		
T1	P	Gewoon van thuis mee.	Follow-up answer		
T1	D	Ja, dus u pint altijd of hoe moet ik me dat voor me zien?	Follow-up question		
T1	M	Ja, en ik laat hem weleens pinnen met dat ik erbij ben.	Follow-up answer		
T2	P	Ja, dat verschilt. Met een betaalpas als het thuisgebracht wordt, maar als ik betalingen doe omdat ik iets koop op dingen, doe ik gewoon met ideaal, met iDeal.	Answer		Explicit w explanation
T2	D	En dat gaat goed?	Follow-up question		
T2	P	Ja. En ik heb ook nog een.	Follow-up answer		
T2	D	Oke, oke. De financiën verder, doet u die? De administratie en de financiën?	Question	Literally	
T2	P	Ja.	Answer		Explicit
T2	D	En dat gaat goed?	Follow-up question		
T2	P	Ja.	Follow-up answer		
T2	P	Alleen de onderlinge verrekening moet ik eerlijk zeggen die doet [naam mantelzorger]. Want dat is een heleboel werk, dus dat doet [naam mantelzorger]. Als ik de boodschappen gedaan heb weet je wel, dan lever ik bij haar dan betaal ik 'm dingen enzovoorts.	Follow-up answer		
T2	P	En dan lever ik de kassabon in en dan zorgt [naam mantelzorger] dat dat verdeeld wordt en betaald. En als ze zich vergist dan hoop ik dat het in haar voordeel is, want het is een hoop werk en dan moet ze er niet slechter van worden.	Follow-up answer		
T3	D	Oh ja, oke. En de financiën, de administratie, de financiën regelen?	Question	Paraphrased	
T3	P	Ja, dat heb ik zelf altijd gedaan.	Answer		Explicit w explanation
T3	P	Maar dat schiet er de laatste tijd ook bij in.	Follow-up answer		
T3	M	Ja, we hebben er gelukkig ook iemand voor die dat	Follow-up answer		
T4	D	Ja. En wie doet de financiën en de administratie?	Question	Literally	
T4	P	Dat doet hij tegenwoordig.	Answer		Implicit
T4	P	Heeft ie nooit gedaan.	Follow-up answer		
T5	D	Nee, nee. En rekeningen betalen, pinnen, doet hij dat?	Question	Suggestive	
T5	M	[buitenlandse taal]. Nee, sinds drie jaar dat hij gaat ook niet zijn kaart gebruiken.	Answer		Implicit
T5	M	Omdat hij zegt ik weet niet.	Follow-up answer		

T5	D	Omdat hij niet weet, met pinpas?	Follow-up question		
T5	M	Uhu. Ja, hij heeft ook niet sinds die drie jaar ook niet helemaal niet gedaan, zeg maar.	Follow-up answer		
T5	D	Oke en wist hij niet hoe hij het moest gebruiken, bijvoorbeeld?	Follow-up question		
T5	M	Ja, hij zei dat kon niet vanwege vergeet ik de code of kan ik verkeerd doen.	Follow-up answer		
T6	D	En de financiën werd ook gedaan toch? Door de? Dus het betalen van de rekeningen?	Question	Suggestive	
T6	P	Nee, dat doet op het ogenblik een vriendin, maar eigenlijk wil ik daar iets anders in.	Answer		Explicit w explanation
T7	M	Hij houdt ook de rekeningen bij [onhoorbare tekst].	Answer		Implicit
T7	D	Oh, ja, de financiën. En dat gaat ook goed?	Question	Following	
T7	M	Dat mag ik hopen.	Answer		Implicit
T8	D	En wie doet de financiën voor u thuis?	Question	Paraphrased	
T8	P	Zelf.	Answer		Explicit
T8	D	Bijvoorbeeld rekeningen betalen en?	Follow-up question		
T8	P	Dat gaat allemaal via de bank.	Follow-up answer		
T8	D	Helemaal goed, en doet u ook wel eens pinnen?	Follow-up question		
T8	P	Ja.	Follow-up answer		
T8	D	Ja, dus dat gaat ook goed? U vergeet de code niet bijvoorbeeld?	Follow-up question		
T8	P	Nee, nee. Soms dan denk ik, wat was het nou, ja, nou dat komt vanzelf wel weer terug.	Follow-up answer		
T8	D	Ja, en financiën heeft u eigenlijk ook niet echt hulp bij he?	Question	Suggestive	
T8	P	Nee.	Answer		Explicit
T9	D	Ja, en, en bijvoorbeeld de financiën? De geldzaken?	Question	Paraphrased	
T9	P	Ja dat, dat loopt allemaal.	Answer		Explicit w explanation
T9	D	Uhu. Ja, en dat gaat goed ook? De financiën?	Follow-up question		
T9	P	Ja.	Follow-up answer		
T10	D	Ja, en bijvoorbeeld de, de financiën, de geldzaken?	Question	Paraphrased	
T10	P	Doe ik ook zonder problemen.	Answer		Implicit
T10	D	Ja, ja en ook met het internetbankieren of gaat dat allemaal goed?	Follow-up question		
T10	P	Dat gaat tot nu toe in ieder geval allemaal goed, ja.	Follow-up answer		
T11	D	Nooit gedaan. Nee, nee. En uw zoon zei al, de, de geldzaken, financiën dat deed u vroeger altijd?	Question	Suggestive	
T11	P	Ja, altijd, he.	Answer		Explicit w explanation
T11	D	Doet u dat nog steeds?	Follow-up question		
T11	P	Nou, nou, niet alleen. Ik ga met mijn dochter of mijn zoon.	Follow-up answer		

T11	M	[onhoorbare tekst] financieel gebied [onhoorbare tekst].	Follow-up answer		
T11	P	Nee, ik bedoel als ik bijvoorbeeld ga pinnen of iets.	Follow-up answer		
T11	M	Ja, dat doe je wel zelf, maar.	Follow-up answer		
T11	P	Maar ik heb alles automatisch, alles gaat automatisch.	Follow-up answer		
T11	D	Uhu. En wat, wat doet uw dochter bij de financiën ? Waar helpt ze u mee?	Follow-up question		
T11	P	Mijn dochter, mijn dochter, ja, die helpt mij bijvoorbeeld als ik ga bij haar lopen, iets voor mij bestellen, of weet je wel, ja, dus.	Follow-up answer		
T12	D	En bijvoorbeeld de financiën thuis, de geldzaken. Regelt u dat zelf of krijgt u daar hulp bij? Dat iemand met u meekijkt?	Question	Paraphrased	
T12	P	De, de, ja, de financiële zaken, die worden door de kinderen gedaan.	Answer		Explicit w explanation
T12	D	Uhu. Ja, want is dat omdat uw man dat altijd bijvoorbeeld gedaan heeft of?	Follow-up question		
T12	P	Ook, maar dat is nogal omvangrijk dus dat, dat doen ze zelf met, wie doet het allemaal? [naam zoon]	Follow-up answer		
T12	D	Ja, dus de zakelijke kant is door de kinderen overgenomen, maar gewoon de dagelijkse financiën voor bij u thuis, de rekeningen betalen.	Follow-up question		
T12	P	Ja, de, ja, de boodschappen, dat soort dingen, dat doe ik ja.	Follow-up answer		

Table 5.22 indicates the geriatric assessment for the category finance, as well as the assumed scores based on transcripts. All values can be found one or more times in these transcripts. As is the tendency with all these categories, the independent score (1) is usually accompanied with the words 'zelf', as well as the words 'yes' and 'good'. Dependent scores (0) are given to patients who fully depend on, in most cases, family members or friends. Mentioning of other people by patients is usually an indicator of either score 0 or 1*. The main difference is that for score 1* patients at least are able to pin and do groceries.

There is one inconsistency between the geriatrician and the transcripts. Patient 11 receives help from his or her children with finances and ordering online, while the doctor evaluated that the patient is independently able to do finances. One possible explanation is that ordering online is not deemed required to be able to do finances.

Table 5.22: Geriatric assessment and transcript scores: Finance

No.	GS	TS	Dif.
Transcript 1	1*	1*	
Transcript 2	1	1	
Transcript 3	1	1	
Transcript 4	0	0	
Transcript 5	0	0	
Transcript 6	0	0	
Transcript 7	1	1	
Transcript 8	1	1	
Transcript 9	1	1	
Transcript 10	1	1	
Transcript 11	1	1*	*
Transcript 12	1*	1*	

5.4 Patterns and conventions in reporting

Doctors report their assessment as a combination of the answers, as well as other factors as explained by prof dr. Yvonne Schoon. It is not uncommon for some parts of the specific anamnesis to already be discussed in previous segments of the CGA. It can be assumed that some items such as transfer and mobility can be assessed by evaluating the patient's entrance, journey to the geriatric department and walking aid. Unless these things are specifically mentioned and thus transcribed, the conversation interpretation algorithm will not be able to assess these items. Doctors typically document their assessment using a checklist, which they can access and update using a tablet.

The scores of the twelve patients who cooperated for this research were analyzed to find the most common and uncommon answers. The findings for the ADL items are presented in Table 5.23. Green cells indicate the most frequent score. Red cells indicate that the score of 3 is not possible for those items. The scores match the explanation of Table 5.1. For mobility, every patient scored three points, meaning that they are all independently able to move, possibly with the aid of a walking cane or other tool. Two patients were dependent of other people while dressing, whereas six patients were independent. All seven patients were able to independently go up or down stairs.
The Lawton index uses a more complex system as explained in chapter 5.1. The results of the IADL index are shown in table 5.24, with the matching explanation available in Table 5.2. There is a bigger variety of values and the patients are more evenly distributed between them.

Table 5.23: Frequency scores ADL

Legend	
(green)	Most frequently answered
(red)	Score unavailable for category

	Score			
Item	**0**	**1**	**2**	**3**
Mobility	0	0	0	9
Dressing	2	0	6	
Stairs	0	0	7	

Table 5.24: Frequency scores IADL

Legend Most frequently answered

Item	**0****	**0***	**0**	**1**
Shopping	3	1	2	4

Item	**0**	**1*****	**1****	**1***	**1**
Housekeeping	2	1	1	3	4

Item	**0**	**1***	**1**
Finances	3	3	6

The final results for Barthel are calculated by adding up all the scores of the individual categories. The final score lies somewhere between 0 and 20. Multiple interpretations are possible depending on the healthcare institute, but one Dutch document explaining the Barthel-index uses the following interpretation (Engelen & Bokhorst, 2016). A score of 20 indicates complete independence. A score between 15 and 19 indicates reasonably or mostly independent. 10 – 14 indicates that partial assistance is required, 5 – 9 indicates a serious dependence on others, and 0 – 4 a full dependence on others. Whatever the interpretation is, it can be an important factor in determining if a patient is able to live independently, or perhaps should move to assisted living. For Lawton the scores of the eight individual categories are added up, meaning that the final score lies between 0 and 8. The asterisks are ignored for the purpose of the numerical assessment.

6. Matching consultation transcripts to geriatric ontologies

In Chapter 6, we will examine the procedure for interpreting geriatric dialogues and the necessary steps for doing so. To begin with, we will outline the distinctions between this process and the general Care2Report conversation interpretation pipeline. Next, we will present the Medical Guideline Ontology of the Barthel and Lawton indices. After that in Section 6.1, we will delve into the process of constructing a narrative from conversations, which is essential for generating triples. Finally, in Section 6.2, we will generate triples from the narratives and provide a visual representation of the knowledge graph's matching with the Medical Guideline Ontology.

To redesign C2R for the purpose of automatically assessing geriatric consultations, the pipeline has to be specified for geriatrics. Figure 6.1 represents the in 4.4 discussed ontological conversation interpretation pipeline, but specified to the Barthel and Lawton indices. The input for the domain ontology learning part are the Barthel and Lawton indices, for which Kemper created a Medical Guideline Ontology (Kemper et al., 2021). In this chapter we will discuss multiple steps in this pipeline. The discussed s

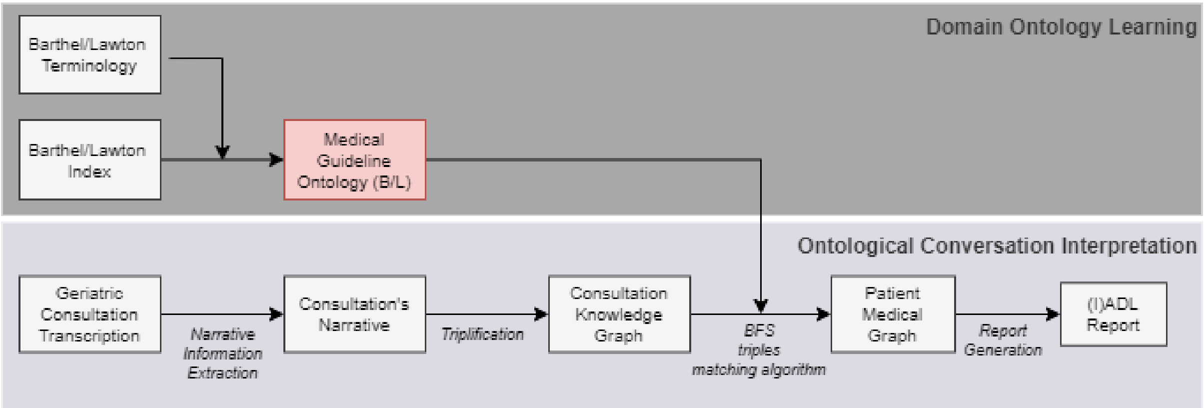


Figure 6.1: Barthel/Lawton ontological conversation interpretation pipeline.

Figure 6.2 shows the Medical Guideline Ontology for all ten Barthel categories. There are some key differences compared to general Medical Guideline Ontologies. First of all, the diagram contains no diagnosis or treatment ontologies, as these are not part of the Barthel and Lawton methods. The anatomic functions are linked to the symptoms. These symptoms (blue) are the ADL categories. Each category has multiple items (observations, green), which indicate the final assessment and are populated while running the triples matching algorithm. We will look at the category mobility as an example. In black we see the anatomical structure and functions of the patient. The patient's body is composed of various anatomic structures, including the head, arms, and legs, which possess specific functions such as memory and activity. These functions can have signs (symptoms), in this instance mobility. Observations then indicate that the patient’s mobility falls within one of the following values or items: dependent, independent with wheelchair, walks with help of 1 person or independent.

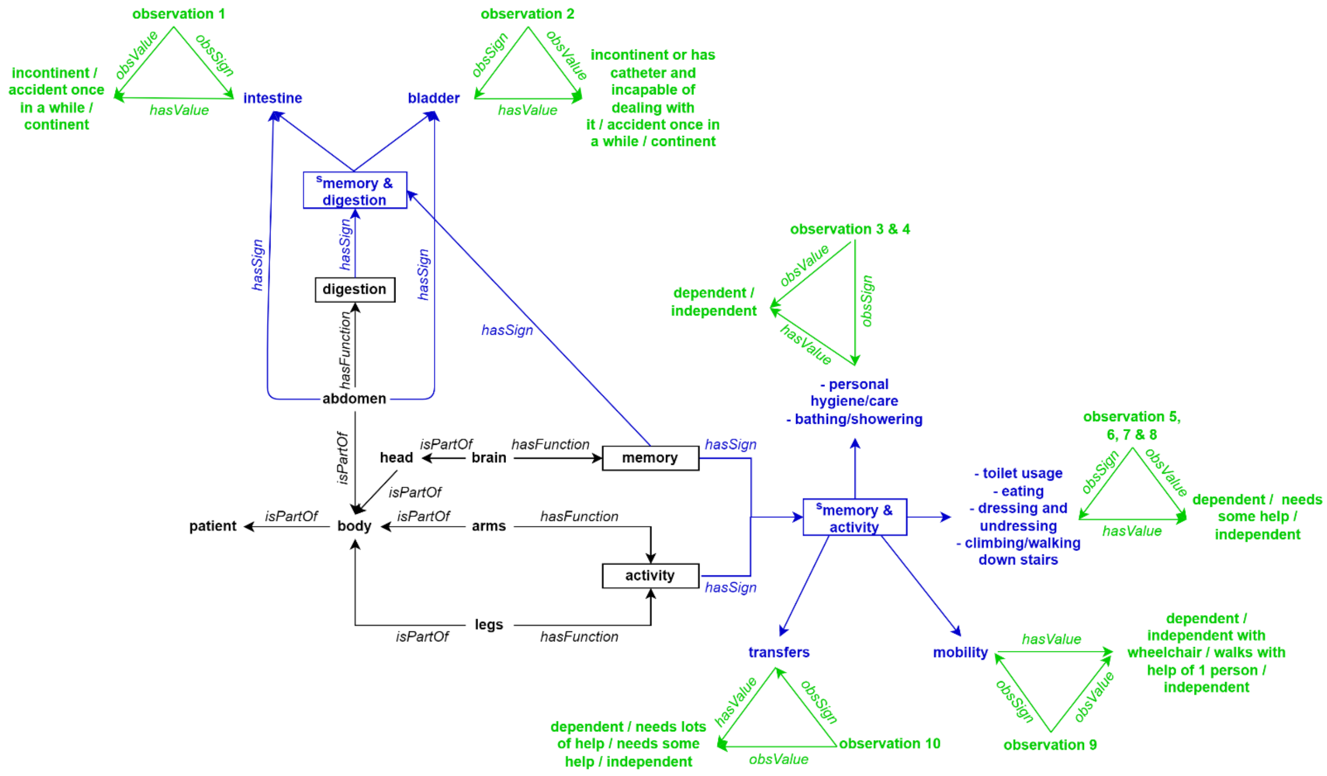


Figure 6.2: Barthel Medical Guideline Ontology (Kemper et al, 2021)

The Lawton list is shown in Figure 6.3 and features similar characteristics, beginning with anatomic structures and corresponding functions. From there, activities are linked to those functions. For example, in the category of shopping, observations by the doctor may lead to the selection of one of four possible values, with the remaining values being excluded during triples matching.

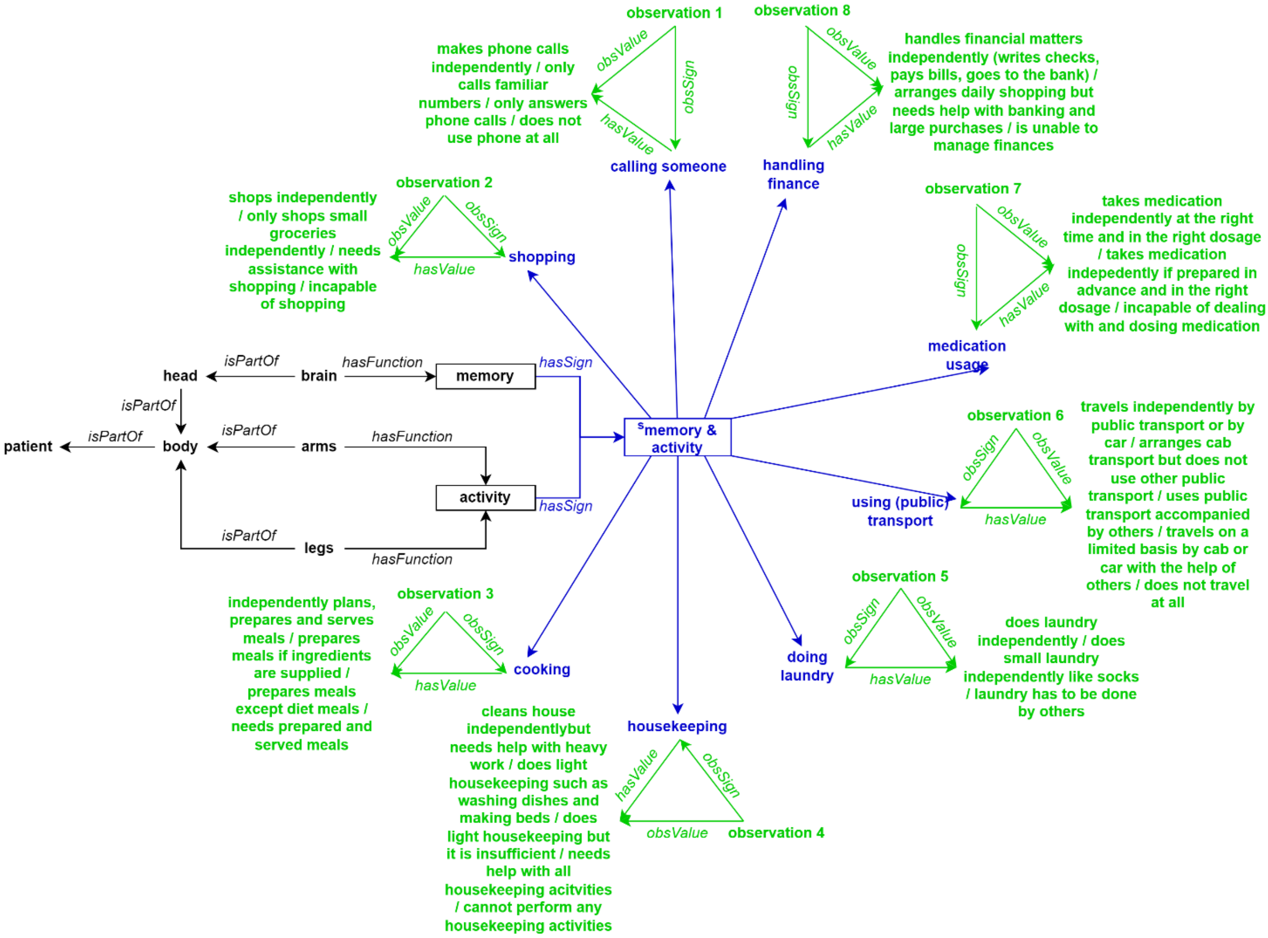


Figure 6.3: Lawton Medical Guideline Ontology (Kemper et al, 2021)

6.1 Narrative Information Extraction

In section 6.1 we will discuss the Narrative Information Extraction process from the Barthel/Lawton ontological conversation interpretation pipeline, which is marked red in Figure 6.4. Both the reason as to why this process is important, as well as its design will be discussed.

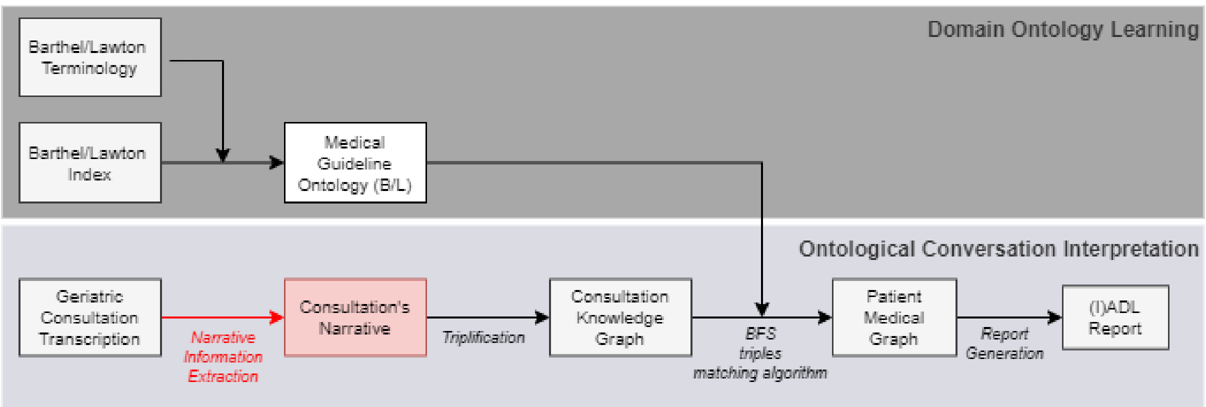


Figure 6.4: Barthel/Lawton ontological conversation interpretation pipeline: Narrative Information Extraction

One of the main problems of summarizing dialogue is the fact that both the doctor's questions and the patient's answer are required to make an accurate statement about the patient. Generating a triple out of 'Yes' or 'No' is not possible, and neither is generating triples out of questions alone.

While monologues can be accurately summarized by matching triples, conversations such as the CGA pose problems, as information resides in multiple speaking turns. There are difficult concepts in natural language, for example negative questions.

"Doctor: You said you are not able to walk stairs correct?

Patient: Yes"

In the example above the patient answers "yes" to the doctor, however this answer should be coded as *unable*. The yes confirms that the patient is not able to perform the action. Generating triples out of questions and answers without processing the data therefore results in a messy output. Furthermore, the duration of (I)ADL consultations can be long, and not all information is required for the final assessment. Before triplification takes place, the transcripts are preprocessed to remove any irrelevant information, as well as transforming them to a format that allows for more accurate triple generation. In theory, preprocessing the triples should reduce the amount of errors during triplification and triples matching. Transcripts contain grammatical errors and large amounts of noise. By correcting errors and removing noise, more consistent and relevant triples will be generated, that are more likely to match with the Medical Guideline Ontology. In actuality, this preprocessing includes the extraction of the narrative from the original dialogue. The narrative is a conversation transformed into a story form, or in this case, into a report-like form. This process is often called narrative information extraction. The goal is to automatically retrieve accurate assessments, but the highest accuracy would be achieved by manually extracting the narrative. A balance should therefore be found by finding the highest accuracy while maintaining a high level of automatization.

6.2 Transcript cleansing and transformations

This section will describe a method for finding a short geriatric narrative out of CGA conversations. To find all possible narrative information extractions steps, one narrative was manually created for each category. These transcript parts were manually picked by including as much conversational variety as possible over those six sections as a representation of the entire dataset. This means that the total sample set includes long dialogue, short dialogue, carers, speaking error etc. Table 6.1 shows the transcript for which each dialogue to narrative transformation was performed.

Table 6.1: dialogue to narrative sample

Category	Transcript ID
Mobility	1
Dressing	11
Stairs	5
Shopping	1
Housekeeping	12
Finance	11

Mobility 1 is used as an example to illustrate finding the narrative information extraction steps. The left text box of figure 6.5 shows the original dialogue, while the right text box shows the desired output. This is the desired output, because it contains all important information required for the ADL scoring while leaving out any irrelevant language. To achieve the desired narrative, parts of the dialogue were either removed or changed (transformations). These transformations were given identifiable names and each unique transformation was listed. All transformations are explained with examples later in this chapter. For mobility 1, the following transformations were identified:

1. Removal of semantically redundant phrases
2. Transition from explicit patient answers to full semantic sentences
3. Rewriting personal pronouns to 3rd person agent
4. Removal of doctor's questions

*“**D:** And walking? Do you walk without aid or do you have a walking stick or walker?* ***P:** No, without.* ***D:** Completely without. And do you ever fall?* ***P:** No.* ***D:** And how goes walking?* ***P:** Goes well.”*	• *Patient walks without aid* • *Patient never falls* • *Patient can walk well*

Figure 6.5: Dialogue and narrative of mobility 1

After identifying all transformations for the six individual dialogues, the transformations were combined into a single list of transformations, which are displayed below but are yet to be ordered:

- Removed transcript [tags]
- Removed semantically incomprehensible or unclear phrases
- Removed (I)ADL irrelevant topics
- Removed semantically redundant phrases
- Corrected grammar and sentence structure
- Removed semantically duplicate sentences
- Transformed explicit/short patient and carer answers to full semantic sentences
- Summarised larger answers
- Rewrote personal pronouns to 3rd person agent (patient)
- Removed of doctor’s questions
- Separated combined sentences into multiple speaking turns
- Rewrote structure for consistency
- Combined patient's and carer's answers into one answer
- Combined patient clarification with original statement

While analysing the above list, some criteria and guidelines were discovered that the final ordered transformation list should adhere to. Generally, two types of transformations were found: transformations that remove dialogue and transformations that modify transformations. To avoid redundant work, all removing transformations should be performed before modifying transformations so that no dialogue is modified that will ultimately be removed. The most important transformation is ‘Transform explicit/short patient and carer answers to full semantic sentences’. Information from question and answers is combined into a single sentence, resulting in having the narrative in a single sentence instead of multiple. This transformation should be done before the doctor’s questions can be removed entirely. Beside these criteria, the chronological order of the six sample dialogue to narrative transformations were used to develop the final ordered list, shown in Table 6.2. The narrative information extraction consists of two main processes. The first process is termed data cleansing, because it comprises of steps that remove assessment-irrelevant data, as well as redundant language use. These transformations are identified as C1 – C6. The second main process is termed dialogue-to-narrative transformation. These transformations focus on extracting the narrative out of both doctor’s questions and patient’s answers to allow for the removal of speaking turns (T1 – T8).

Table 6.2: (I)ADL conversations – Transcript cleansing and transformations

Main process	Transformations
Data cleansing	C1. transcript [tags] removal C2. (I)ADL irrelevant topics removal C3. Semantically incomprehensible or unclear phrases removal C4. Semantically duplicate sentences removal C5. Semantically redundant phrases removal C6. Grammar and sentence structure correction
dialogue-to-narrative transformations	T1. Large answer summarisation T2. Explicit/short patient and carer's answers transformation to full semantic sentences T3. Personal pronouns rewriting to 3rd person agent (patient) T4. Doctor's speaking turns removal T5. Combined sentence separation into multiple sentences T6. Patient's and carer's answers combining into one answer T7. Patient's clarification combining with the original statement T8. Sentence rewriting to a consistent sentence structure

The following two sections (6.2.1 and 6.2.2) will provide examples of each transformation as performed on the transcripts. The description of each transformation will be presented in text, followed by an example drawn from the transcripts. The left column shows the transformation number, which corresponds to the numbers in Table 6.2. The middle column shows the original dialogue, while the final column indicates how the transformation changes the dialogue. Red text indicates that part of the dialogue will be removed, while blue text indicates the part of the dialogue that is changed.

6.2.1 Transcript cleansing

C1. transcript [tags] removal: This step involves the removal of all transcript [tags]. During transcribing, certain audio parts could not be transcribed due to audio quality issues, or because someone speaks a foreign language. These audio parts are indicated by tags describing the reason there is no transcription there. During transformation 1 these tags are removed.

C 1	**D**: Ja, maar met veel moeite. **M**: Veel moeilijk, ja. **D**: Ja, ja. En de trap af kost ook moeite. Trap op? **M**: [buitenlandse taal]. **P**: [buitenlandse taal]. **M**: Ja.	• ***D:*** *Ja, maar met veel moeite.* • ***M:*** *Veel moeilijk, ja.* • ***D:*** *Ja, ja. En de trap af kost ook moeite. Trap op?* • ***M:*** *Ja.*

C2. (I)ADL irrelevant topics removal: This removes any parts of the transcripts that is irrelevant for the (I)ADL assessment. Naturally, conversations sometimes drifts away to other topics. These parts are removed so that they do not have to be transformed later in the process.

C 2	**M:** Hij handelt in munten en zo en dat gaat goed. **D:** Oh oke, in bitcoins of zo iets? **M:** Nee. **P:** Nee, nee, dan zou ik nu schatrijk geweest zijn, als ik daar op tijd mee begonnen was. **M:** Twee euro munten die hij dan gaat verkopen. **D:** Oke, oke. De financiën verder, doet u die? De administratie en de financiën?	**D:** Oke, oke. De financiën verder, doet u die? De administratie en de financiën?

C3. Semantically incomprehensible or unclear phrases removal: This involves the removal of sentence fragments that are semantically incomprehensible due to factors such as missing syntax or context, and as such cannot be transformed into a narrative.

C 3	**D**: Ja, wat lastig. **M**: Uit handen neem. **D:** Ja, maar u vergeet dus soms dan producten? **P:** Ja.	**D**: Ja, wat lastig. **D:** Ja, maar u vergeet dus soms dan producten? **P:** Ja.

C4. Semantically duplicate sentences removal: During this step, sentence (fragments) are removed which are semantically identical to previously mentioned sentence (fragments). It is redundant to have multiple identical statements in the narrative.

C 4	**M:** Maar doet ie wel alles, dat zelfstandig, maar. **D:** Ja, maar met veel moeite. **M:** Veel moeilijk, ja. **D:** Ja, ja. En de trap af kost ook moeite. Trap op? **M:** Ja.	**M:** Maar doet ie wel alles, dat zelfstandig, maar. **D:** Ja, maar met veel moeite. **M:** Veel moeilijk, ja. **D:** Trap op? **M:** Ja.

C5. Semantically redundant phrases removal: This step removes phrases that are semantically meaningless. These phrases primarily include interjections, hesitation markers (uhm), stop words and certain expressions of feelings.

C 5	**M:** Hmm. **D:** Voor corona en voor? **M:** Niet echt, maar wel ja, voor sowieso ouderen. **D:** Precies.	**D:** Voor corona en voor? **M:** Niet echt, maar voor sowieso ouderen. **D:** Precies.

C6. Grammar and sentence structure correction: Errors in grammar and sentence structure are corrected in this step. People do not always speak in full or grammatically correct sentences. This step improves the use of language to make the narrative extraction easier.

C 6	**M:** Niet echt, maar voor sowieso ouderen. **D:** Precies. **M:** doet ie wel alles, dat zelfstandig	**M:** Niet echt, maar voor ouderen sowieso. **D:** Precies. **M:** Hij doet wel alles zelfstandig.

6.2.2 Narrative transformations

Following the process of data cleansing, the narrative transformations are performed. Once more, the explanations and examples of all transformations will be demonstrated.

T1. Large answer summarization: This consists of summarising long answers and stories to the semantic core. This reduces the size of the eventual narrative and filters out irrelevant information.

T 1	**M:** Alleen laatst met die broodjes. Dat ging mis bij de bakker. Moest hij de bestelling ophalen. Dat is een bruin brood en twee zakken broodjes, maar ik had iets waardoor we nog meer broodjes nodig hadden dus extra broodjes….	**M:** Door een vergissing bij de bakker heeft hij teveel broodjes gekocht.

T2. Explicit/short patient and carer's answers transformation to full semantic sentences: This is one of the most important dialogue-to-narrative transformations. As identified at the beginning of this chapter, one of the main problems of summarising CGA dialogue is the fact that information resides in multiple speaking turns. Transformation 8 addresses this by enriching the explicit answer of the patient by using the context of the doctor's question. In practice this means that 'Yes', 'No' and other short answers are rewritten to full sentences including the pronoun or subject of the patient and the activity the doctor asked about. If we take the earlier example of:

"Doctor: You said you are not able to walk stairs correct?

Patient: Yes"

We can transform this answer to a statement which can be accurately triplified by including the doctor's question:

"Doctor: You said you are not able to walk stairs correct?

Patient: I am unable to walk stairs"

T 2	**D:** Ook het wassen, aankleden, doet u allemaal zelf? **P:** Ja	**D:** Ook het wassen, aankleden, doet u allemaal zelf? **P:** Ik was mezelf en ik kleed mezelf aan.

T3. Personal pronouns rewriting to 3rd person agent (patient): This transformation aims to rewrite all personal pronouns so that it is clear who the subject of the statement is, in most cases the patient. In some cases a carer may have answered for the patient, in which case the pronoun can cause confusion and should still be replaced by 'patient'.

T3	**D:** Ook het wassen, aankleden, doet u allemaal zelf? **P:** Ik was mezelf en ik kleed mezelf aan	**D:** Ook het wassen, aankleden, doet u allemaal zelf? **P:** Patiënt wast en kleedt zelf.

T4. Doctor's speaking turns removal: Now that the patient's answers are enriched with the doctor's questions, the doctor's questions have become redundant and can therefore be removed entirely.

T4	**D:** Ook het wassen, aankleden, doet u allemaal zelf? **P:** Patiënt wast en kleedt zelf.	**P:** Patiënt wast en kleedt zelf.

T5. Combined sentence separation into multiple sentences: This transformation separates combined sentences so that during triplification separated activities are not seen as one activity. "Patient can shower and dress himself". Triplification in this case would yield: {Patient, can, shower and dress himself}, while the preferred output would be: {Patient, can, shower himself} and {Patient, can, dress himself}.

T5	**P:** Patiënt wast en kleedt zichzelf.	Patiënt wast zelf Patiënt kleedt zelf

T6. Patient's and carer's answers combining into one answer: This combines answers from multiple speaking turns into one complete answer. In some cases a carer might provide additional information to a patient's original answer. In that case the answers are combined to provide the complete answer.

T6	**P:** Nee, ik bedoel als ik bijvoorbeeld ga pinnen of iets. **M:** Ja, dat doe je wel zelf, maar.	Patiënt pint zelf

T7. Patient's clarification combining with the original statement: This is nearly identical to Transformation 6, except that it includes the explanation once the doctor has enquired about additional information.

T7	**P:** Patiënt kan één boodschap doen, maar vergeet meer boodschappen. **D:** Één boodschap of een product? **M:** Eén product.	Patiënt kan één product kopen, maar vergeet meerdere producten.

T8. Sentence rewriting to a consistent sentence structure rewrites: During this transformation all narrative sentences are rewritten into an almost identical format to improve the accuracy of triplification and decrease the chance of ambiguity.

T 8

• Patiënt gaat met moeite met de trap naar beneden. • Voor corona ging Patiënt niet met moeite met de trap naar beneden. • Patiënt doet wel alles zelfstandig.	• Patiënt gaat met moeite met de trap naar beneden. • Patiënt ging voor corona niet met moeite met de trap naar beneden • Patiënt doet wel alles zelfstandig.

6.3 Dialogue to narrative: worked out example

Now that the narrative information extraction transformations have been established, all 12 transcripts will be transformed to narratives following the above steps. The following section will include a few examples of the complete transformation process. It's important to note that not all steps will be required for each individual transcription. The following Dutch example is drawn from transcript 7 and pertains to the category shopping. The following example represents a typical conversation during the anamnesis process, and includes various elements that may complicate the process of generating a narrative, such as transcription tags, explicit and implicit responses, language errors, and the presence of a carer.

The textboxes of Figure 6.6 show this example, with each text box representing a transformation. Each text box will display changes made to the text using red font to indicate deleted sections and blue font to indicate changed sections. The text boxes are organized chronologically, with the first box displaying the initial state and the first transformation, the second box displaying the second transformation, and so on. Transformations C1 – C6 and T2 through T5 were necessary to form a narrative for this transcript. Transformations T1, T6, T7, and T8 were not included in forming the narrative.

C1. transcript [tags] removal: Eliminated inaudible text [tags] that were inserted during transcribing.

D: Het, de boodschappen doet u die zelf?
M: Hij doet de boodschappen.
P: Ik doe alle boodschappen.
D: Oke.
M: [onhoorbare tekst]. Je moet voorkomen [onhoorbare tekst].
D: Ja, dus.
M: [onhoorbare tekst].
D: Ja, oke. Dus u doet de boodschappen, en hoe gaat dat?
P: Prima.
D: Maakt u een lijstje op.
P: Lijstje en ik heb gelukkig al mijn boodschappen bijna op, doe ik tussen half acht en acht uur.
D: Oh, ja.
P: En dan is er niemand in de zaak die mij dwars kan zitten want, ja, op files en op een dag bij Lidl of weet ik het wat, is het best druk. Daar ontkom je bijna niet aan, dat je dus die anderhalf meter nog kunt handhaven.
D: Maar het lukt u goed om te vinden wat u zoekt en.
P: Ik weet waar iets ligt, ja.

Figure 6.6: Transformation C1 - Removed transcript [tags]

C2. (I)ADL irrelevant topics removal: One section in the text discusses how crowded stores can get being the reasoning why the patient shops between 7:30 and 8:00. This text would not appear in the specific anamnesis report and is therefore deemed irrelevant and removed in **C2**.

D: Het, de boodschappen doet u die zelf?
M: Hij doet de boodschappen.
P: Ik doe alle boodschappen.
D: Oke.
M: Je moet voorkomen
D: Ja, dus.
D: Ja, oke. Dus u doet de boodschappen, en hoe gaat dat?
P: Prima.
D: Maakt u een lijstje op.
P: Lijstje en ik heb gelukkig al mijn boodschappen bijna op, doe ik tussen half acht en acht uur.
D: Oh, ja.
P: En dan is er niemand in de zaak die mij dwars kan zitten want, ja, op files en op een dag bij Lidl of weet ik het wat, is het best druk. Daar ontkom je bijna niet aan, dat je dus die anderhalf meter nog kunt handhaven.
D: Maar het lukt u goed om te vinden wat u zoekt en.
P: Ik weet waar iets ligt, ja.

Figure 6.6: Transformation C2 - Removed (I)ADL irrelevant topics

C3. Semantically incomprehensible or unclear phrases removal: The phrase *"je moet voorkomen"* is incomplete, likely due to the missing inaudible parts, and therefore removed in **C3.** The response *"ja, dus"* from the doctor is also considered obsolete as a result.

D: Het, de boodschappen doet u die zelf?
M: Hij doet de boodschappen.
P: Ik doe alle boodschappen.
D: Oke.
M: Je moet voorkomen
D: Ja, dus.
D: Ja, oke. Dus u doet de boodschappen, en hoe gaat dat?
P: Prima.
D: Maakt u een lijstje op.
P: Lijstje en ik heb gelukkig al mijn boodschappen bijna op, doe ik tussen half acht en acht uur.
D: Oh, ja.
D: Maar het lukt u goed om te vinden wat u zoekt en.
P: Ik weet waar iets ligt, ja.

Figure 6.6: Transformation C3 - removed semantically incomprehensible or unclear phrases

C4. Semantically duplicate sentences removal: This involves the removal of repetitions. The patient repeats the answer provided by their caregiver, while the doctor repeats her own response.

D: Het, de boodschappen doet u die zelf?
M: Hij doet de boodschappen.
P: Ik doe alle boodschappen.
D: Oke.
D: Ja, oke. Dus u doet de boodschappen, en hoe gaat dat?
P: Prima.
D: Maakt u een lijstje op.
P: Lijstje en ik heb gelukkig al mijn boodschappen bijna op, doe ik tussen half acht en acht uur.
D: Oh, ja.
D: Maar het lukt u goed om te vinden wat u zoekt en.
P: Ik weet waar iets ligt, ja.

Figure 6.6: Transformation C4 - removed semantically duplicate sentences

C5. Semantically redundant phrases removal: There is not much redundancy in this transcript, but **C5** involves the removal of a few semantically redundant words.

D: Het, de boodschappen doet u die zelf?
M: Hij doet de boodschappen.
D: Oke.
D: Dus u doet de boodschappen, en hoe gaat dat?
P: Prima.
D: Maakt u een lijstje op.
P: Lijstje en ik heb gelukkig al mijn boodschappen bijna op, doe ik tussen half acht en acht uur.
D: Oh, ja.
D: Maar het lukt u goed om te vinden wat u zoekt en.
P: Ik weet waar iets ligt, ja.

Figure 6.6: Transformation C5 - removed semantically redundant phrases

C6. Grammar and sentence structure correction: *"ik heb gelukkig al mijn boodschappen bijna op, doe ik tussen half acht en acht uur"*. This sentence is grammatically incorrect and is corrected during **C6** now that the meaning can still be derived from the context.

D: De boodschappen doet u die zelf?
M: Hij doet de boodschappen.
D: Oke.
D: Dus u doet de boodschappen, en hoe gaat dat?
P: Prima.
D: Maakt u een lijstje op.
P: Lijstje en ik heb gelukkig al mijn boodschappen bijna op, doe ik tussen half acht en acht uur.
D: Oh, ja.
D: Maar het lukt u goed om te vinden wat u zoekt.
P: Ik weet waar iets ligt.

Figure 6.6: Transformation C6 - correct grammar and sentence structure

T2. Explicit/short patient and carer's answers transformation to full semantic sentences: This example includes two explicit or short answers, one of which one is embedded in a compound sentence. **T2** rewrites these sentences as full sentences comprising a subject, verb and object in order to maintain the semantics of the narrative.

D: Doet u de boodschappen zelf?
M: Hij doet de boodschappen.
D: Oke.
D: Dus u doet de boodschappen, en hoe gaat dat?
P: Prima
D: Maakt u een lijstje op?
P: Lijstje en ik doe gelukkig bijna al mijn boodschappen tussen half acht en acht uur.
D: Oh, ja.
D: Maar het lukt u goed om te vinden wat u zoekt?
P: Ik weet waar iets ligt.

Figure 6.6: Transformation T2 - Transformed explicit/short patient and carer's answers to full semantic sentences

T3. Personal pronouns rewriting to 3rd person agent (patient): The personal pronouns are revised to refer to the subject of this report, namely the patient, enabling the removal of the doctor's text **(T4. Doctor's speaking turns removal)**, as her information has all been incorporated into the answers.

D: Doet u de boodschappen zelf?
M: Hij doet de boodschappen.
D: Oke.
D: Dus u doet de boodschappen, en hoe gaat dat?
P: Boodschappen doen gaat prima.
D: Maakt u een lijstje op?
P: Ik maak een lijstje en ik doe gelukkig bijna al mijn boodschappen tussen half acht en acht uur.
D: Oh, ja.
D: Maar het lukt u goed om te vinden wat u zoekt?
P: Ik weet waar iets ligt.

Figure 6.6: Transformation T3 – Rewrote personal pronouns to 3rd person agent (Patiënt)

D: Doet u de boodschappen zelf?
M: Patiënt doet de boodschappen.
D: Oke.
D: Dus u doet de boodschappen, en hoe gaat dat?
P: Boodschappen doen gaat prima.
D: Maakt u een lijstje op?
P: Patiënt maakt een lijstje en patiënt doe gelukkig bijna al zijn boodschappen tussen half acht en acht uur.
D: Oh, ja.
D: Maar het lukt u goed om te vinden wat u zoekt?
P: Patiënt weet waar iets ligt.

Figure 6.6: Transformation T4 – Removed doctor's speaking turns

T5. Combined sentence separation into multiple sentences: Finally, there is one more interesting transformation to perform. One combined answer essentially includes two separate statements, which are separated during **T5**. Because no answers require any more clarification and the structure seems consistent, the last three steps can be skipped. The speaking turn indicators are therefore removed and we are left with the narrative.

M: Patiënt doet de boodschappen.
P: Boodschappen doen gaat prima.
P: Patiënt maakt een lijstje. patiënt doe gelukkig bijna al zijn boodschappen tussen half acht en acht uur.
P: Patiënt weet waar iets ligt.

Figure 6.6: Transformation T5 – Separated combined sentences into multiple sentences

- Patiënt doet de boodschappen.
- Boodschappen doen gaat prima.
- Patiënt maakt een lijstje.
- Patiënt doe gelukkig bijna al zijn boodschappen tussen half acht en acht uur.
- Patiënt weet waar iets ligt.

Figure 6.6: Narrative of shopping transcript 7

6.4: Conclusion of transformations and worked out example

During this chapter we developed six data cleansing transformations and eight narrative transformations that allowed us to transform dialogues into narratives, so that the output can be used for the generation of triples. These transformations were illustrated by performing the steps on various fragments of the transcripts. The full dialogue to narrative transformations were then performed on the shopping category of transcript 7 in chapter 6.3. The results of the final transformation will be discussed here.
Figure 6.7 and 6.8 present both the dialogue and narrative of the shopping category of transcript 7. The first thing that can be noticed is the reduction in size. The total amount of lines is reduced from 15 speaking turns consisting of multiple sentences, to five concise sentences. Additionally, The total amount of words has been reduced from 144 to 30 words, which is a decrease of 79%. The speaking turns have been removed, because the subject of each sentence is mentioned in the narrative. Knowledge triples have the form of subject-predicate-object. The subject should therefore be easy to identify when generating triples. Furthermore, the concise nature of the narratives already look similar to the knowledge triples. For example: *"Patient doet de boodschappen"* can result in the triple: {*subject*: "Patient", *Predicate:* "doet", *Object:* "de boodschappen"}. When comparing Figure 6.7 and 6.8, the left out information can be identified. For example, information about files (traffic jams) and the patient's preferred supermarket (Lidl) are filtered out, as this is irrelevant to the doctor's assessment.

D: Het, de boodschappen doet u die zelf?
M: Hij doet de boodschappen.
P: Ik doe alle boodschappen.
D: Oke.
M: [onhoorbare tekst]. Je moet voorkomen [onhoorbare tekst].
D: Ja, dus.
M: [onhoorbare tekst].
D: Ja, oke. Dus u doet de boodschappen, en hoe gaat dat?
P: Prima.
D: Maakt u een lijstje op.
P: Lijstje en ik heb gelukkig al mijn boodschappen bijna op, doe ik tussen half acht en acht uur.
D: Oh, ja.
P: En dan is er niemand in de zaak die mij dwars kan zitten want, ja, op files en op een dag bij Lidl of weet ik het wat, is het best druk. Daar ontkom je bijna niet aan, dat je dus die anderhalf meter nog kunt handhaven.
D: Maar het lukt u goed om te vinden wat u zoekt en.
P: Ik weet waar iets ligt, ja.

Figure 6.7: Dialogue of shopping transcript 7

- Patiënt doet de boodschappen.
- Boodschappen doen gaat prima.
- Patiënt maakt een lijstje.
- Patiënt doe gelukkig bijna al zijn boodschappen tussen half acht en acht uur.
- Patiënt weet waar iets ligt.

Figure 6.8: Narrative of shopping transcript 7

6.5 Ontology and triples matching

Having access to the narratives now enables the generation of knowledge triples. Given the current limited availability of triplification and NLP software in languages other than English, all transcripts were translated into English prior to the creation of English narratives. Both Dutch and English dialogues and narratives are available at this moment. During the translation process, the aim was to not only accurately translate the words, but also to preserve the original meaning and expressions, as well as to translate language errors into comparable errors in English. The Barthel and Lawton Medical Guideline Ontologies are cropped to a shorter version as to only include our six selected categories. The condensed MGO's are displayed in Figures 6.9 and 6.10.

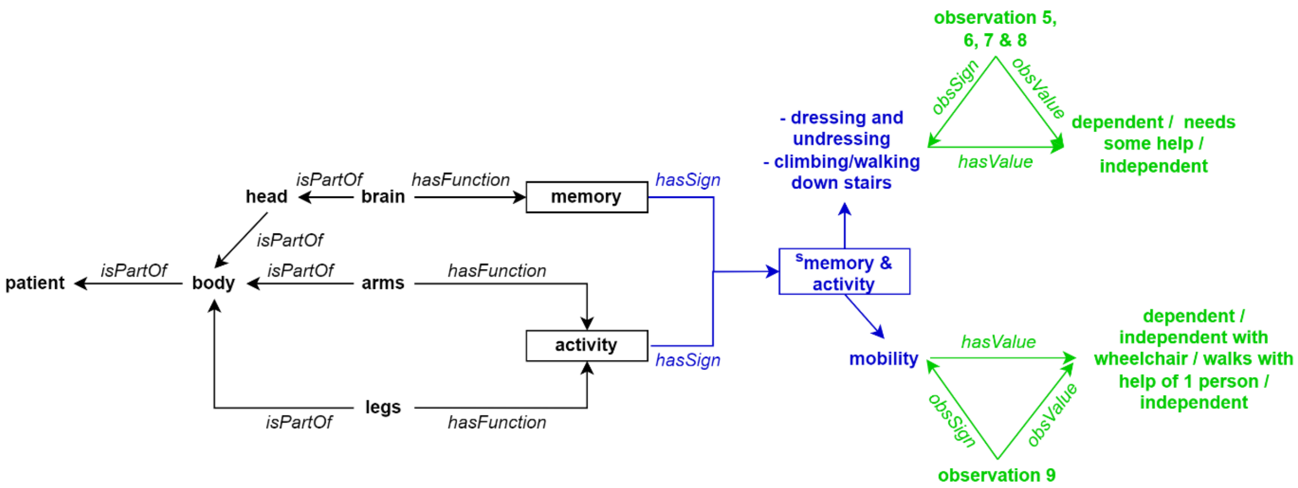


Figure 6.9: MGO ADL restricted to Mobility, Dressing and Stairs

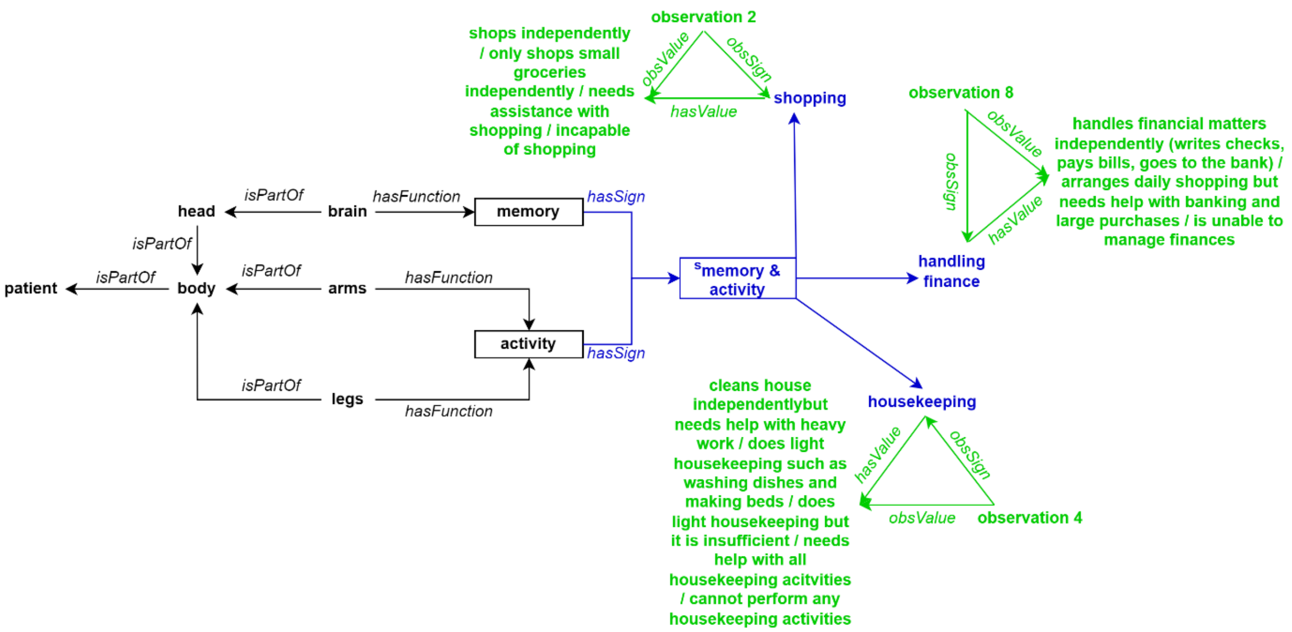


Figure 6.10: MGO IADL restricted to Shopping, housekeeping and finances

6.5.1: Triple generation

During section 6.5.1 we will discuss the triple generation (triplification) using the generated narratives as input. The current step in the ontological conversation interpretation pipeline is highlighted in Figure 6.11.

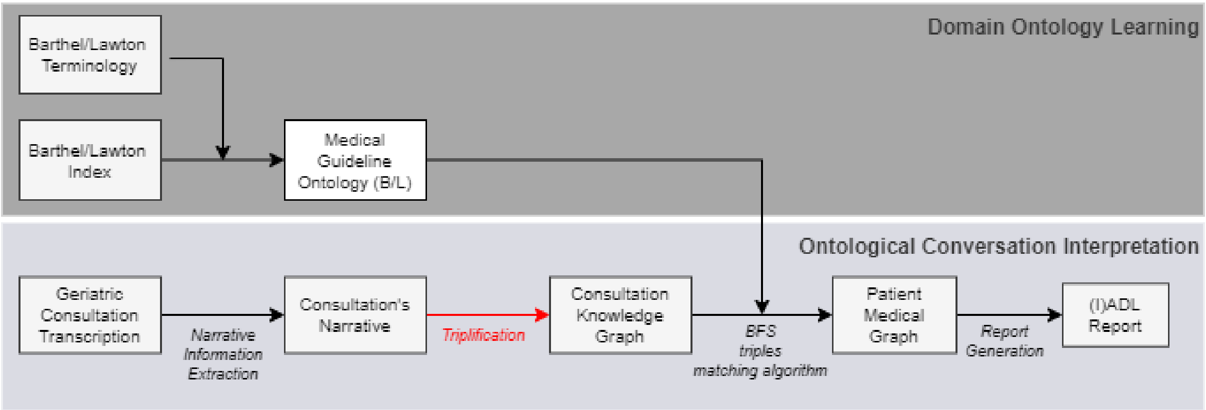


Figure 6.11: Barthel/Lawton ontological conversation interpretation pipeline: Triplification

A Python wrapper for Stanford openIE is used for the triplification process. Stanford openIE is a relatively easy to use open IE system. Open Information Extraction (Open IE) systems aim to extract tuples from text containing relations stated in text. While tools such as FRED transforms the text into knowledge triples suitable for RDF and OWL, open IE simply links two elements from the text with a relation. Stanford openIE specifically divides a sentence into entailed coherent clauses (Angeli, Premkumar & Manning, 2015). This is done by traversing a dependency parse tree and predicting for each node if it should yield a clause. These clauses are then transformed into smaller sentences. From these smaller sentences the relationship, object and subject are identified. Stanford openIE's precision goes down with a higher recall (Stanovsky & Dagan, 2016), so post-processing is required to remove false positives. Code-snippet 6.1 displays the most basic code used to generate triples. As the code-snippet indicates, not many lines of code are required for it to run. There are multiple options which can be adjusted to change output, but these are not required for this purpose. The code opens a txt file, annotates the corpus and prints the triples.

```
// import the StanfordOpenIE library
from openie import StanfordOpenIE

// Opens the txt file containing the narrative
with StanfordOpenIE(properties=properties) as client:
    with open('PathToFile', encoding='utf8') as r:

// Reads the files and removes linebreaks.
corpus = r.read().replace('\n', ' ').replace('\r', '')

// Annotates the corpus and prints the triples
    triples_corpus = client.annotate(corpus[0:5000])
    print('Corpus: %s [...].' % corpus[0:80])
    print('Found %s triples in the corpus.' % len(triples_corpus))
    for triple in triples_corpus:
        print(triple)
```

Code-snippet 6.1: Stanford openIE code

In order to demonstrate the output of Stanford OpenIE's triplification process, we utilized the English narrative of transcript 1 as input. As shown in Table 6.4, the input sentences and the resulting output triples are presented. From the 20 input sentences, a total of 33 triples were generated. Of these triples, 15 were identified as true positives, while the remaining 18 were either duplicates or were deemed grammatically and semantically nonsensical and were therefore excluded from the table. It is worth noting that true negatives do not pose a problem in this context, as they can be filtered out during the triples matching algorithm. However, six sentences did not yield any triples, and the reason for this is unclear, as the sentences in question were relatively simple. Modifying the Stanford OpenIE properties did not alter the results. Calculating the exact recall can be challenging, because an input sentence can yield multiple correct triples. For the purposes of validating the Stanford OpenIE's triplification software for our case study, recall is defined as the percentage of sentences that yield at least one or more semantically complete triple. The triplification recall for the narrative of patient 1 is calculated to be 70%. Although the recall and precision values are not optimal, the narratives generally produce clean output with minimal noise.

Table 6.4: Stanford openIE Triplification output patient 1

Nr.	**Input**	**Output**		
		Subject	**Predicate**	**Object**
1	Patient walks without a walker.	Patient	walks without	walker
2	Patient never falls.			
3	Patient can walk well.			
4	Patient can climb stairs.	Patient	can climb	stairs
5	Patient has a stairwell.	Patient	has	stairwell
6	Patient never takes the elevator.			
7	Patient writes groceries on a note.	Patient	writes groceries on	note
8	Patient forgets products.	Patient	forgets	products
9	Patient can buy a product.	Patient	can buy	product
10	Patient cannot purchase multiple products.			
11	Patient does not buy any products twice.			
12	Patient does housekeeping.	Patient	does	housekeeping
13	Patient does everything himself.	Patient	does everything	himself
14	Patient pays in cash.	Patient	pays in	cash
15	Patient brings cash from home.	Patient	brings cash from	home
16	Carer always uses card.	Carer	always uses	card
17	Carer sometimes lets patient pay with card.	Carer	sometimes lets	patient pay
		Carer	lets	patient pay with card
18	Carer sometimes has to tell patient what the PIN is.			
19	Patient sometimes forgets debit card PIN.	Patient	sometimes forgets	debit card pin
20	Carer has always done finances and administration.	Carer	has always done	finances

To research whether Stanford OpenIE is an acceptable library to generate triples out of narratives, all 12 transcripts have been triplified. Table 6.5 demonstrates the recall for each transcript, as well as the overall recall. The left column contains the transcript numbers. The next column shows the amount of sentences which yielded satisfactory triples. The third column indicates the total amount of sentences the patient's narrative have. The final column presents the recall by dividing the number of correct sentences by the total number of sentences. The overall output recall of Stanford openIE triplification using our data is 67%. There is a large variance between transcripts, which cannot completely be explained logically. The main category which seemed to yield few triples was the category of *dressing*. Perhaps the combination of "dressing and undressing" in those sentences confused the triplication software as to which predicate to take. In the current form this would not be sufficient for the triplification process, as the risk of losing the information of entire categories is too large. Care2Report is still looking for the best software to use for triplification.

Table 6.5: Recall of Stanford OpenIE triplification results

Trans-cript Nr.	Sentences with correct triples	Total sentences	Recall
1	14	20	70%
2	15	26	58%
3	13	16	81%
4	12	20	60%
5	8	14	57%
6	8	9	89%
7	6	10	60%
8	9	16	56%
9	6	8	75%
10	2	2	100%
11	7	11	63%
12	8	10	80%
Total	108	162	67%

6.5.2: Triple matching

The final components of the pipeline discussed in this chapter are related to the triples matching algorithm. This process is marked red in Figure 6.12. During the triples matching, the Medical Guideline Ontology is populated with the triples from the Consultation Knowledge Graph to create the Patient Medical Graph.

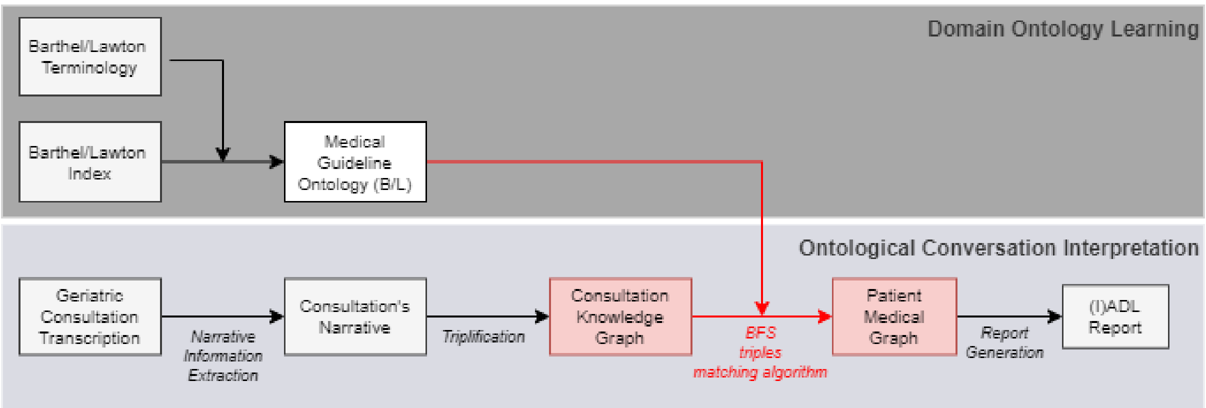


Figure 6.12: Barthel/Lawton ontological conversation interpretation pipeline: Triple matching

The in the previous section generated triples will not be matched to the ontologies as it is beyond the scope of the project. Nonetheless, an example visualization of the triple matching algorithm is provided in Figure 6.13 to illustrate the conversation interpretation phase of the Care2Report pipeline. This figure shows on the top the barthel MGO of our three selected ADL categories, and on the bottom the in FRED generated knowledge graph of the following three sentences:

- Patient can walk well.
- Patient can walk stairs.
- Patient does housekeeping.

The black line, and highlighted black textbox represent the anatomical structure of the patient, who serves as a subject in the knowledge graph. Blue lines and textboxes represent words that can be connected to a category, and indicate which category the triple is about. Green lines and textboxes represent words that specify which item should be selected and indicate the patient’s dependency.

While walking through the breadth first search algorithm, predicates such as *walking* are identified. The algorithm recognizes this predicate as one that belongs to the mobility category. The knowledge graph shows that walk hasQuality of well. The knowledge graph also indicates that *walk* is associated with the quality of *well*, which is interpreted as a positive attribute and can therefore be linked to the item independent. Similarly, *stairs* indicates that that triple represents the category of walking stairs. FRED has transformed the word “can” in the sentence “Patient can walk stairs” to the modality *possible.* The word *possible* indicates that the patient is indeed able to walk stairs, so the item independent will be selected. It should be noted that the sentence about housekeeping is included in the knowledge graph as well. This sentence would naturally be matched to the IADL Medical Guideline Ontology.

While this was a relatively straightforward example, there may be numerous phrases that suggest a person's ability to perform a task independently. Future research can provide answers to the most effective method of recognizing these various dependency indicators.

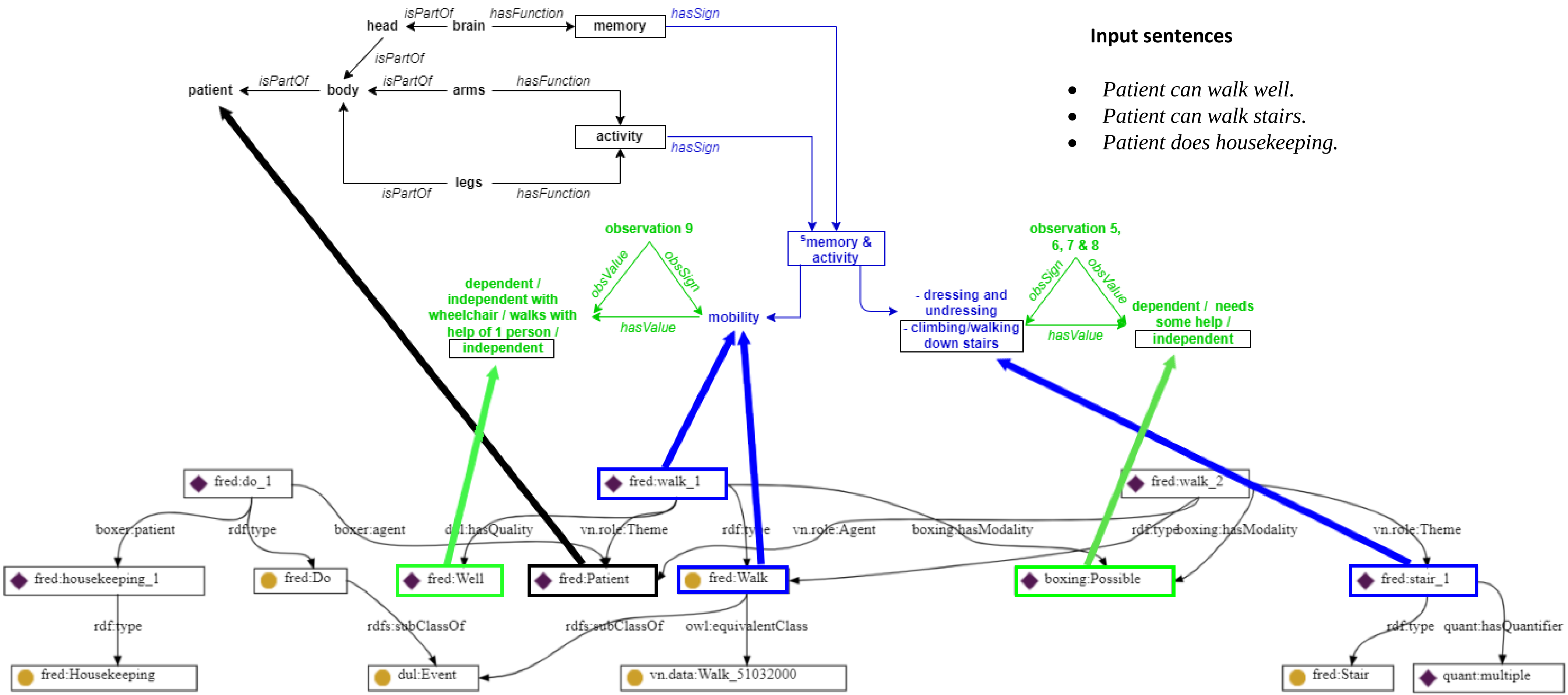


Figure 6.13: Visualization of the triples matching algorithm

7. Automatic geriatric narrative information extraction

Chapter 7 will focus on the automation of geriatric narrative information extraction. In section 7.1, various approaches to this problem are discussed, and the chosen approach will be discussed. In section 7.2, the use of dependency parsing as a means to extract the syntax of sentences will be discussed. Section 7.3 will delve into the implementation of SimpleNLG as a means to generate narratives. Finally 7.4 will discuss the final pipeline and results. The automated process described in this chapter relates to the Narrative Information Extraction activity in our pipeline, which is marked red in Figure 7.1.

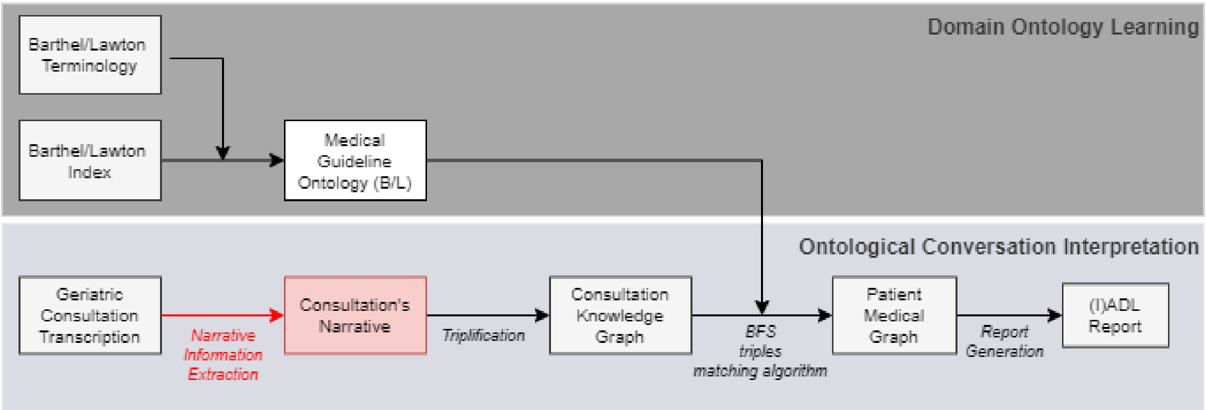


Figure 7.1: Barthel/Lawton ontological conversation interpretation pipeline: Automated Narrative Information Extraction

7.1 Automating dialogue to narrative transformations

There are multiple options to consider when designing a system for extracting the narrative from conversations. One approach is to utilize linguistic techniques, while another option is to focus on semantic analysis. One approach is through the use of machine learning techniques, while another approach is through the use of rule-based methods. While machine learning is the more prevalent method in modern times, it presents some challenges in this particular context. One issue is that a sufficient training dataset is not available, as only twelve transcripts are at our disposal. Additionally, there are currently no publicly available machine learning models specifically designed for this task in the domain of medical conversations. Due to these limitations and challenges, a rule-based design is chosen for this case study. Given the limitations of rule-based systems, a linguistic approach in this context is preferred over a semantic approach, as syntactic structures inherently involve the use of rules.

The aim of creating the narrative is to extract the essential information from the conversations, allowing the removal of speaking turns and unnecessary phrases. One of the key transformations identified in chapter 6 was rewriting explicit answers by including the doctor’s question in the answer. Therefore, the automated design should be able to create statements that include both the doctor's question and the patient's response. To generate a narrative out of a conversation, the following steps have to be taken. The bold text highlights the section in which the specific step is discussed.

1. **Design setup (7.1.1) -** Identify which syntactic elements of conversational sentences generally contain essential information.
2. **Design setup (7.1.1) -** Find or design a system that is able to identify those syntactic elements in sentences.
3. **Dependency parse (7.2) -** Extract the words from sentences that match the identified key syntactic elements.
4. **Sentence realiser (7.3) -** Use those words to realise new sentences that together form a narrative.

7.1.1 Design setup

The first step is to identify which syntactical elements are most key to the narrative of a sentence. This will be primarily done with trial and error, assuming that the most basic form of a sentence includes a subject, a predicate and an object. More information on which syntactic components are used for the generation of sentences is described in the two upcoming sections.

The second step is to find a way to identify the required syntactic components in sentences. Many options exist, but the common linguistic technique is to use a dependency parser and part-of-speech tagger. SpaCy is the free open-source library used for this task, as it is generally accurate and efficient. 7.2 will explain the code written to identify and extract the key syntactic components from our transcript sentences, thereby also completing step 3. Finally the in 4.2 introduced tool SimpleNLG will be utilized to realise new sentences.

7.2 Dependency parse

To identify and extract predetermined part-of-speech and dependencies from sentences, the SpaCy library is used. SpaCy is compatible with Windows, macOS and Linux, with the Windows system being used in this particular instance. SpaCy was build using Python, and was installed using package manager pip. SpaCy's neural network models are available in over 23 languages, including Dutch, English, most prevalent European languages and Chinese. To integrate SpaCy into the overall pipeline, the English model was installed. During installation it is possible to specify the pipeline for accuracy or for efficiency. adding *_trf* to the download optimises the pipeline for accuracy, whereas *_sm* optimises for efficiency (Code-snippet 7.1). Since we are working with relatively few data at the moment, the accuracy pipeline is chosen.

```
$ python -m spacy download en_core_web_trf   # Accuracy
$ python -m spacy download en_core_web_sm   # Efficiency
```

Code-snippet 7.1: SpaCy installation

To use the SpaCy library, the module should be imported using *import spacy*. If one wants to visualize the techniques (e.g. create a parsetree), display should also be imported as demonstrated in Code-snippet 7.2. In our pipeline the json module is also imported, which allows for the storing and transferring of data using JSON files.

```
import spacy
from spacy import displacy
import json
```

Code-snippet 7.2: Import libraries

As there appeared to be no documentation of which unique POS tags and dependencies SpaCy is able to extract, a small script was created to find all unique values. Code-snippet 7.3 demonstrates how this is done for the POS tags. Dependencies are found in the exact same way, but by replacing each *pos* in the code by *dep*. The code first loads the language model from SpaCy. It then creates a list which will include all unique POS tags. Then a function is created which goes over each sentence in the text and creates a *doc* object from the nlp model for each sentence. This *doc* object is a container which allows for linguistic annotations. The *nlp* function takes the sentence as input and performs the various nlp techniques, such as the POS tagger. Then for each word in this doc object, it stores the POS tag in a variable, and if that tag is not yet present in the list, it adds the POS tag to the list. The full code for this process, including that of the dependency finder can be found in Appendix 7.1. This also includes how an entire folder can be processed and how the function is called.

```
nlp = spacy.load("en_core_web_trf")  # Loads the SpaCy English model
pos_wordtypes = []  # Create a list for all possible part-of-speech tags

def unique_pos(text): # Create a function for finding the POS tags
    for sentence in text:  # For each sentence in the text
        doc = nlp(sentence) # Creates a pre-processed Doc object from the nlp model
        for words in doc: # Goes over each word in the Doc object
            pos_wordtype = words.pos_ # For each token stores tag in the variable
            if pos_wordtype not in pos_wordtypes:
                pos_wordtypes.append(pos_wordtype) # If the token is not yet in the list,
it unique pos_tag.

    return "POS: " + str(pos_wordtypes)
```

Code-snippet 7.3: Find unique part-of-speech tags

Subsequently, we conducted an analysis to determine the frequency of occurrence of specific parts of speech and dependencies within the transcripts. Specifically, we calculated the percentage of sentences in which these linguistic features were present. Tables 7.1 and 7.2 presents a summary of the most frequently occurring and noteworthy part-of-speech and dependency. In total, there are 570 speaking turns in the transcripts. The second column indicates the total amount of speaking turns the syntactic element can be found in. The final column shows the relative presence compared to the 570 total speaking turns. The first thing important to note is that even the most important syntactic elements are present in only just over 50% of the speaking turns, indicating that many speaking turns consist of only single words or short phrases. Additionally, the parser identified that many sentences include interjection, which are often short spontaneous expressions or reactions, as well as filler words, which in almost all cases are not essential for the narrative. Besides from the obvious appearance of nouns and verbs, auxiliary verbs also have a relatively high presence in the transcripts. Finally, pronouns are prevalent and the main challenge here is to identify whether the pronouns are talking about the patient or something or someone else.

Table 7.1: POS occurence

POS	Sentences (out of 570)	Percentage
Pronoun	332	58%
Interjection	332	58%
Verb	321	56%
Noun	310	54%
Adverb	244	43%
Auxiliary	221	39%

Table 7.2: Dependency occurence

POS	Sentences (out of 570)	Percentage
Root	570	100%
Subject	312	55%
Adverb modification	240	42%
Object	216	37%
Auxiliary	185	32%

Now that the various possible syntactic elements are identified, the transcripts can be parsed and the desired words can be extracted. To achieve this, the language model of SpaCy is again used to run by each token (word), and identify the corresponding part-of-speech and dependency. Code-snippet 7.4 demonstrates the code for this task. This first iteration is the most basic version, and only extracts parsed verbs and objects. The subject is always assumed to be the patient. The algorithm will go over each sentence in the file, and find all tokens in that sentence (the *doc* object). First, lists are created for all required linguistic elements. Then the algorithm runs through each token, and identifies whether it is a verb (*VERB*), direct object (*dobj*) or indirect object (*pobj*). If that is the case, then that token is added to the earlier created lists.

```python
import spacy
import json

nlp = spacy.load("en_core_web_trf")  # Loads SpaCy’s language model
transcript = open("path to json file") # Assigns the content of our JSON file with tran-
scripts to a variable called transcript
text = json.load(transcript) # Converts JSON file to Python object

output = []
id = 0 #
for sentence in text["Transcriptie"]:
    print(sentence)
    doc = nlp(sentence)
    id = id + 1
    verbs = []
    dobj = []
    pobj = []

# for each sentence and for each token in those sentences, if the token = verb, add it to
the list of verbs. Same for objects.
    for token in doc:
        if(token.pos_ == "VERB"):
            verbs.append(token.text.lower())
        if(token.dep_ == "dobj"):
            dobj.append(token.text.lower())
        if(token.dep_ == "pobj"):
            pobj.append(token.text.lower())

# Adds the tokens to the output. Separated per sentence.
output.append({"id" : id, "sentence" :sentence, "verbs": verbs, "dobj" :dobj, "pobj"
:pobj})

json_string = json.dumps(output)

with open('name outputfile', 'w') as outfile:
    outfile.write(json_string)
```

Code-snippet 7.4: Initial dependency parse code

When the algorithm has run through all sentences, it will create the output showing the verbs and objects of each sentence. The format of the output JSON file is shown in Figure 7.2. The figure only shows one sentence, but the complete file includes all sentences below each other.


```json
{
  "id": 1,
  "sentence": "And walking? Do you walk without aid or do you have a walking stick or walker?",
  "verb": [
    "walking",
    "walk",
    "have",
    "walking"
  ],
  "dobj": [
    "stick"
  ],
  "pobj": [
    "aid"
  ]
},
```


Figure 7.2: SpaCy parse output

7.3 Sentence realiser

Section 7.3 will discuss the realising of sentences using SimpleNLG. In 7.3.1 the base SimpleNLG setup will be discussed. The treatment design was done by completing various cycles of designing the dependency parser and sentence realiser, analysing the results, and again updating the design (iterations). Each individual iteration and their results were discussed in the various subsections. 7.3.2 discusses the first iteration two iterations, which involve a primary functionality and data cleansing. Iteration three (7.3.3) involves determining the root of a sentence. Iteration four includes various changes to root and subject and handling. Finally in iteration five adpositions and modifiers are added to the code.

7.3.1 Basic SimpleNLG setup

Now that the code is written which allows us to extract required words based on specified linguistic elements, new summarised sentences can be realised using SimpleNLG. A C# port was used for SimpleNLG to integrate the code in a Swagger API environment. SimpleNLG at its core is relatively easy to understand. In the most basic form, a sentence will be created with a subject, verb and object. The first step is to create a Lexicon, NLGFactory and a realiser using the three statements found in Code-snippet 7.5. A lexicon includes the words and information about those words. The NLGFactory is an object which includes methods used to specify sentence structure. Finally, the realiser is responsible for generating sentences as text. An instance of a sentence (SPhraseSpec) is created using the createClause method. All part of speech and dependencies are structured and composed in this instance using the various NLGFactory methods. After creating the instance, all components the sentence should include are specified. In Code-snippet 7.1 strings are given directly to the setSubject, setVerb and setObject methods. Finally, the realiser will take the components and combine them according to the rules we have set up, while making sure that the syntax and morphology is correct. With the components set in this example, the output will be: “A man walks with his dog”. This is not surprising, as it is a relatively straightforward sentence in which the input was literally specified, but many more functionalities can be utilised.

```java
import simplenlg.framework.*;
import simplenlg.lexicon.*;
import simplenlg.realiser.english.*;
import simplenlg.phrasespec.*;
import simplenlg.features.*;


public class TestMain {

  public static void main(String[] args) {

    Lexicon lexicon = Lexicon.getDefaultLexicon(); // Simple lexicon included in simpleNLG
    NLGFactory nlgFactory = new NLGFactory(lexicon); // Object which creates
    SimpleNLG structures
    Realiser realiser = new Realiser(lexicon); // Object which transforms SimpleNLG
    structures into text

    SPhraseSpec p = nlgFactory.createClause();  // Creates an instance of the class
    SPhraseSpec

    p.setSubject("A man"); // sets the specified subject
    p.setVerb("walks"); // sets the specified verb
    p.setObject("with his dog"); // sets the specified object

    String output2 = realiser.realiseSentence(p); // calls on the sentence realiser
    System.out.println(output2);
 }
}
```

Code-snippet 7.5: Basic SimpleNLG sentence creation (in Java)

The goal is to automatically realise sentences using the output of SpaCy's POS and dependency parse. A pipeline was created to automatically use the various elements of the parsed JSON files as input for these clauses such as setSubject and setVerb. To get a base model running, the first goal was to create sentences using only a subject, verb and object, but the goal is to eventually increase the amount of components to be able to generate all the relevant narrative sentences. A C# program was created that takes the JSON properties, assigns them variable names and inputs those in the SimpleNLG library. A class called ParsedSentence takes each individual property of the parsed JSON file, and assigns a list variable to it. These lists contain all subjects, verbs and objects of each sentence. Furthermore, a series of string variables are created in SentenceConfiguration, which take the first of these ParsedSentence list variables and stores them into these single string variables. Looking at Code-snippet 7.6, all subjects in the JSON file (marked with nsubj) of a single sentence are stored in a list called **Subjects**. A method *Convert()* will then take the first subject from the list **Subjects** and assign that to a variable called **Subject**. This is done for all JSON properties so that each sentence has one subject, one verb and one object.

```
public class ParsedSentence
   {
     [JsonPropertyName("id")] public int Id { get; set; }

     [JsonPropertyName("sentence")] public string? OriginalSentence { get; set; }

     [JsonPropertyName("nsubj")] public List<string>? Subjects { get; set; }

     [JsonPropertyName("verb")] public List<string>? Verbs { get; set; }

     [JsonPropertyName("dobj")] public List<string>? DirectObjects { get; set; }
   }
```

Code-snippet 7.6: Assigning variables to Json properties (C#)

The variables created in the previous paragraph will now be used as input in the sPhraseSpec methods. Notice the similarities between Code-snippet 7.7 and Code-snippet 7.5. Again, the nlgFactory, as well as the instance called sPhraseSpec are created at the beginning of the CreateSentence() method. If the **Subject**, **Verb**, and **Object** variable are not empty, it will assign these as properties of the setSubject(), setVerb() and setObject() methods. With the right components set the realiser is then able to generate the sentences.

```
public string CreateSentence(SentenceConfiguration config, FeatureConfiguration
featureConfig)
  {
     var nlgFactory = _simpleNlgRepository.GetNlgFactory();
     if (config.IsEmpty()) return string.Empty;
     var sPhraseSpec = nlgFactory.createClause();

     if (config.Subject != null) sPhraseSpec.setSubject(config.Subject);
     if (config.DirectObject != null) sPhraseSpec.setObject(config.DirectObject);
     if (config.IndirectObject != null)
sPhraseSpec.setIndirectObject(config.IndirectObject);
     if (config.Root != null) sPhraseSpec.setVerb(config.Predicate);


    foreach (var feature in _mapper.Map<List<KeyValuePair<string,
object>>>(featureConfig))
    {
        sPhraseSpec.setFeature(feature.Key, feature.Value);
    }

    return _simpleNlgRepository.GetRealiser().realiseSentence(sPhraseSpec);
  }
}
```

Code-snippet 7.7: Realising sentences using parsed sentences (C#)

7.3.2 Iteration 1 & 2: Base & Data cleansing

The output file can be downloaded in SwaggerUI as a JSON file, an example of which can be seen in Figure 7.3. Each sentence includes the original sentence, as well as the output sentence generated by SimpleNLG. Immediately it is noticeable that the generated sentences either are not semantically logical, or do not conform with the meaning of the original sentence.

```
[
  {
    "originalSentence": "And walking? Do you walk without aid or do you have a walking stick or walker?",
    "output": "Patient walks aid stick."
  },
  {
    "originalSentence": "No, without.",
    "output": ""
  },
  {
    "originalSentence": "Completely without. And do you ever fall?",
    "output": "Patient falls."
  },
```


Figure 7.3: SimpleNLG output downloaded from SwaggerUI as JSON file.

To analyse the results, the JSON output of all twelve transcripts were combined, and converted to an Excel file, a fraction of which can be seen in Figure 7.4. The file includes the transcript numbers of the sentences' source, as well as the original sentence and the output sentence. As the figure shows, not every input sentence also generates an output sentence. In most cases, the reason for this is that the input sentence is a single word, or a combination of words that do not include a verb or an object. Even though the subject is always set as *patient*, SimpleNLG probably does not generate sentences including only a subject. Additionally, it is noticeable that the output cells that are not empty, are often not the preferred output. *"Patient example"* is obviously not the correct narrative of the original sentence. The only correct output sentence in this example is *"Patient cycles"* as it accurately reflects the input sentence of *"I also cycle"*. These results already indicate that at the moment, the simpler the original sentence, the more likely it is the algorithm will produce good output.

Transcript	originalSentence	output
10	Yes, and for example, the finances, the financial matters?	Patient example.
10	I also do without any problems.	Patient does problem.
10	Yes, yes and also with online banking or is that all going well?	Patient goes banking.
10	So far it's all going well.	Patient goes.
10	Yes, yes, okay.	
11	Yourself, yes. And do you also do all the washing, dressing yourself?	Patient dresss washing.
11	Yes Yes.	
11	Yes.	
11	Yeah move, I gotta move for the blood, blood.	Patient moves blood.
11	flow.	
11	Yes.	
11	flow. And I also cycle.	Patient cycles.
11	Yes.	
11	Cycling. I have a bicycle in living room, not upstairs. I have it in the living room, so I car	Patient lives room bicycle.
11	A trainer?	
11	Yes, such a trainer.	

Figure 7.4: Fraction of Excel file of SimpleNLG output

To more accurately assess the validity of the current implementation, the output sentences were scored using a simple point system of incorrect (-), partially correct (o) and correct (+). Incorrect suggests that at least two or more words in the generated sentences would need to be altered in order for them to be considered accurate. Partially correct means that one or two words should be changed, the position of two words need to be switched, or that morphology has to change in order for it to be considered correct. A correct sentence correctly reflects the narrative of the original sentence. As mentioned above, some sentences did not result in an output sentence. If the original sentences were short statements or expressions without verbs, they were removed from the Excel file. Figure 7.5 presents a segment of the cleaned data file and includes the tags for the correctness assessment. Sentence 10 (*"Patient leg"*) is an example of incorrect output (-). Multiple things have to be changed in order for this output to make sense. This does highlight the problem of a lack of context when using this method. Without having the sentence this one is referring to, creating a narrative is difficult. An example of partially correct output is line 6. The only word missing here is the modifier *for* in front of the word blood. Finally, line 7 does result in the desired output.

Sentence nr	Transcript	originalSentence	output	Correctness	Cat
1	10	Yes, and for example, the finances, the financial matters?	Patient example.	-	
2	10	I also do without any problems.	Patient does problem.	-	
3	10	Yes, yes and also with online banking or is that all going well?	Patient goes banking.	o	
4	10	So far it's all going well.	Patient goes.	-	
5	11	Yourself, yes. And do you also do all the washing, dressing yourself?	Patient dresss washing.	-	
6	11	Yeah move, I gotta move for the blood, blood.	Patient moves blood.	o	
7	11	flow. And I also cycle.	Patient cycles.	+	
8	11	Cycling. I have a bicycle in living room, not upstairs. I have it in the li	Patient lives room bicycle.	-	
9	11	To stay active.	Patient stays.	-	
10	11	Yes, for the legs.	Patient leg.	-	
11	11	Yes Yes Yes. And you do your shopping, you say well you will go in th	Patient does neighborhood shopping.	o	
12	11	Yes, at the beginning of corona. When my daughter says, mama you	Patient does beginning shopping.	o	

Figure 7.5: SimpleNLG output including correctness scale

Prior to analyzing the transcripts output it was evident that the data has to be cleaned, before the SimpleNLG output is generated. Not every input sentence should generate a narrative. Some sentences are irrelevant, or only contain confirmations or expressions. As a result, these sentences were eliminated from the JSON files that served as input for the dependency parse code. This reduced the amount of input sentences from 582 to 224 sentences. Ideally, this step could be automated by removing all sentences consisting solely of interjections, for example. However, it would still be necessary to address irrelevant sentences, which cannot be automatically removed. After data cleaning, parsing and realising sentences, the following results were obtained (Table 7.3). The majority of sentences (63%) produced insufficient output sentences. About a quarter of the sentences were partially correct, and 14% of the sentences yielded positive results.

Table 7.3: Output correctness after first iteration

Tag	Sentences	%
-	140	63
o	52	23
+	32	14
Total	224	100

7.3.3 Iteration 3: Sentence root determination

To gain more insight into what causes the incorrect output, the Excel file was enriched by assessing the (presence of) various syntactic components. A hundred sentences were analyzed to see whether all the required syntactic elements of the input sentence appeared in the output. A fragment of the file is shown in Figure 7.5. As explained in section 7.3.1, currently our algorithm picks the first out of a list for each dependency or POS. If there are 5 verbs in a sentence, the algorithm picks the first one, since at the moment we are unable to identify the most important verbs. Therefore, the new columns serve as binary indicators of whether a wrong syntactic element was selected (if present at all) in the generated sentences. A manual comparison was performed between the original sentences and output sentences to see which words were missing. If a linguistic element is missing or incorrect, that cell is marked as *False*. The last three columns assess the difficulty of the original sentence, and are therefore marked with *True* if that is the case. Combined and complex sentences often include multiple of the same syntactic elements and are as a result harder to transform. Additionally, complex sentences often feature more diverse syntactic elements, such as adpositions and compliments, which are not implemented in the current iteration. The first step was to try and improve the selection of the right verb. Out of the 100 tested sentences, at least 30 sentences featured either the incorrect verb, or did not include a verb at all.

Line nr	Tr.	originalSentence	output	Corectness	Right verb	Right subject	Right object	Has modifier/compliment	Has verb	Has adposition	Right morphology	Complex sentence	Combined sentence	Multiple sentences
1	1	And walking? Do you walk without aid or do you have a walking stick or	Patient walks aid stick.	o						FALSE			TRUE	
2	1	Completely without. And do you ever fall?	Patient falls.	+										
3	1	And how goes walking?	Patient goes.	-		FALSE								
4	1	Can you climb stairs?	Patient climbs stair.	+										
5	1	Do you have stairs in the house?	Patient has house stair.	o				FALSE		FALSE				
6	1	Yes. Stairwell. I actually never take the elevator.	Patient takes elevator.	+										
7	1	No, so you do that too and that also works?	Patient works that.	-	FALSE								TRUE	
8	1	I write on a note.	Patient writes note.	o				FALSE						
9	1	Will it still go wrong?	Patient goes.	-		FALSE								
10	1	One grocery goes, but two no longer.	Patient goes.	-		FALSE							TRUE	
11	1	One grocery as in one product?	Patient product.	-		FALSE			FALSE					
12	1	Yes, but you do sometimes forget products?	Patient forgets product.	o				FALSE						
13	1	And do you buy any things double?	Patient buys thing.	o				FALSE						

Figure 7.6: Output including causes of incorrect input

To improve the handling of verbs, the most important verb in a sentence should be identified. The most efficient way to do this is to find the predicate. In theory a predicate can consist of multiple verbs, but in this case it is considered the verb that interacts with the subject and object. It is the verb that has most dependencies with other words. Although SpaCy's dependency parse does not directly identify predicates, Table 7.2 identified that every sentence has a *ROOT*. This ROOT does not show up in SpaCy's parse trees, but can still be found by either looking at the parse tree's dependency arrows, or by simply specifying SpaCy to give the ROOT of a sentence. Code-snippet 7.8 demonstrates the addition to the current parser (upper code) and the SimpleNLG code (lower). An if-statement has been added to add the identified *ROOT* to a list. This root is then inserted into the SimpleNLG realiser as the verb.

```
root = []
for token in doc:
    if(token.dep_ == "ROOT"):
        root.append(token.text.lower())
```

```
if (config.Root != null) sPhraseSpec.setVerb(config.Root);
```

Code-snippet 7.8: Code augmented with ROOT

7.3.4 Iteration 4: Sentence root and subject changes

In iteration 4, two more changes were made to both the parser and realiser. First of all, it became apparent that the Root in SpaCy does not always have to be a verb. Furthermore, SpaCy identifies two types of verbs: *verb* and *aux* (auxiliary verbs). If-statements were added to the code that identified if the root is also a verb or an auxiliary. If that is the case than the token is stored in the variable *pred.* If the root is neither of the two verb forms, it is stored as *root*. The changes to both the parser and realiser can be found in Code-snippet 7.9. The complete code can be found in Appendix 7.2.

```
        # SET ROOT
        # If statement to find and set root if root is not a predicate
        if token.dep_ == "ROOT" and (token.pos_ != "AUX" and token.pos_ != "VERB"):
            root.append(token.text.lower())
        # If statement to find and set predicate
        if token.dep_ == "ROOT" and (token.pos_ == "VERB" or token.pos_ == "AUX"):
            pred.append(token.text.lower())
```

```
if (config.Predicate != null) sPhraseSpec.setVerb(config.Predicate);
```

Code-snippet 7.9: Root to predicate changes. Parser (upper code), Realiser (lower code)

The second change relates to the subject handling. Until now, the subject was consistently set as the patient, but it is also possible for the subject to be other individuals or objects. Various rules were added to the parser to more accurately identify the right subject. The implementation of these rules are presented in Code-snippet 7.10. First of all, it is important to separate personal pronouns (I, you, he, she) from non-personal pronouns. A list of the non-personal pronouns was added to the code. The first if-statement tries to find the subject (*nsubj* or *nsubjpass*) and checks whether that subject does not have the pronoun part-of-speech. If it is indeed a subject, but not a pronoun, the token is stored as the subject. The logic behind this is that if the subject is not a pronoun, it is likely to be an object, a family member or a name. In that case the subject is clearly not the patient, and the subject should therefore be whatever SpaCy found as the subject. The second rule checks whether the token is both a subject and a personal pronoun. In this case it is assumed that the sentence is talking about the patient, because the patient is the subject of the entire conversation. This will not be the case for all situations, but for a vast majority it will be correct. If the subject is a personal pronoun, the subject variable will be set to "patient". The third rule checks whether the subject has a non-personal pronoun, as specified in the earlier created list. Pronouns such as *this*, *that*, *those* are obviously not related to the patient, and are as a result stored as the object of the sentence. Finally, if SpaCy does not find a subject dependency at all, then the subject will default to "patient" as well. The complete parsing code is available in Appendix 7.2.

```
non_personal_pronouns = ["this", "that", "these", "those", "it", "they", "who", "what",
"which"]

 for token in doc:
     # SET SUBJECT
     # If the word is a subject and not a pronoun. Word == subject
     if (token.dep_ == "nsubj" or token.dep_ == "nsubjpass")  and token.pos_ != "PRON":
           nsubj.append(token.text.lower())
     # If the word is a subject and a personal pronoun. Subject == patient
     if(token.dep_ == "nsubj" or token.dep_ == "nsubjpass") and token.pos_ == "PRON" and
     token.text.lower() not in non_personal_pronouns:
           nsubj.append("patient")
     # If the token is a subject and pronoun, but NOT a personal pronoun. Word == subject
     if(token.dep_ == "nsubj" or token.dep_ == "nsubjpass") and token.text.lower() in
     non_personal_pronouns:
           nsubj.append(token.text.lower())

    # If there is no subject, default is "patient"
    if not nsubj:
        nsubj.append("patient")
```

Code-snippet 7.10: Subject addition

After this iteration, a sample of the hundred sentences were assessed on correctness (Table 7.4). Compared to the results after the first iteration, the amount of incorrect sentences has decreased by nearly 20%. The amount of partially correct sentences has gone up by nearly 20%. Finally, the percentage of correct sentences has slightly gone down. This can be explained by the fact that a smaller sample was used of a hundred sentences, instead of all 233 sentences. However, results have improved in the sense that more than half of the sentences are at least partially correct.

Table 7.4: Iteration 4 results

Tag	Sentences	%
-	45	45
o	44	44
+	11	11
Total	100	100

7.3.5 Iteration 5: Objects, adpositions and modifiers

The final iteration alters the way objects are being handled. Additionally, it adds complexity and details to the sentences by adding modifiers and adpositions. The realiser currently adds the direct object and prepositional object after the verbs, without consideration of the right order. In natural language, the direct object is often placed before the prepositional object. For example: "*The man has stairs in his house*". In the current implementation however, the prepositional object is often placed before the direct object, resulting in output such as: *"Man has house stairs"*. To better realise sentences such as the previous example, various changes were made by adding adpositions, as well as forcing the direct object to always be placed in front of the prepositional object. To allow for these changes, we added adpositions, adverbs and adjectives to the parser (Code-snippet 7.11) Additionally, a negate-check has been added to the parser, which can identify whether a sentence is negated.

```
adverb = []
adjective = []
adposition = []
negated = False

for token in doc:
    if token.dep_ == "neg":
        negated = True
    # SET ADVERB
    if token.pos_ == "ADV":
        adverb.append(token.text.lower())
    # SET ADJECTIVE
    if token.pos_ == "ADJ":
        adjective.append(token.text.lower())
    # SET ADPOSITION
    if token.pos_ == "ADP":
        adposition.append(token.text.lower())
```

Code-snippet 7.11: Adjective and modifier addition to dependency parser

The realizer has been extended to include the addition of the new rules (Code-snippet 7.12). The setSubject and setVerb methods have remained unchanged, however the object now utilize a CoordinatedPhrase object. This CoordinatedPhrase allows two components to be combined, and given coordinates so that a specified order can be achieved. First, the direct object is added to the CoordinatedPhrase using addCoordinate(). After that, a PrepositionPhrase is created, which includes both the preposition, as well as the prepositional object (PObject). Finally, the PrepositionPhrase is added as the second coordinate to CoordinatedPhrase objects. setFeature("CONJUNCTION", null) assures the word "and" will not be placed between the two objects. The last two if statements add adjectives and adverbs to the sentences. The implementation of these modifiers however has not worked perfectly as the results will suggest.

```
public string CreateSentence(SentenceConfiguration config, FeatureConfiguration
featureConfig)
{
   var nlgFactory = _simpleNlgRepository.GetNlgFactory();
   if (config.IsEmpty()) return string.Empty;
   var sPhraseSpec = nlgFactory.createClause();
   var objects = nlgFactory.createCoordinatedPhrase();

   if (config.Subject != null) sPhraseSpec.setSubject(config.Subject);
   if (config.Predicate != null) sPhraseSpec.setVerb(config.Predicate);

   if (config.DirectObject != null) objects.addCoordinate(config.DirectObject);
   if (config.PObject != null)
   {
     var PObjectPP = nlgFactory.createPrepositionPhrase();
     PObjectPP.addComplement(config.PObject);
     if (config.Adposition != null) PObjectPP.setPreposition(config.Adposition);
            objects.addCoordinate(PObjectPP);
   }
        sPhraseSpec.setObject(objects);
        objects.setFeature("CONJUNCTION", null);

   if (config.Adjective != null) sPhraseSpec.setFeature("TENSE", config.Adjective
+ "ed");
   if (config.Adverb != null) sPhraseSpec.setFeature("MODAL", config.Adverb);
```
Code-snippet 7.12: Object, adposition an modifier changes to the realiser

To test whether the inclusion of modifiers (adjectives and adverbs) improve the results, a side by side comparison for one of the transcripts has been performed. The results of this comparison are shown in Figure 7.7. A similar correctness assessment has been performed as for the previous iterations, however this time incorrect output is displayed with a red filled cell, partially correct with a yellow fill and correct with a green fill. Additionally, zero points are awarded for incorrect sentences, half a point for partially correct sentences and a full point for correct sentences. From the scores presented in the two output columns, it can be concluded that there is overall minimal difference in accuracy. Some sentences are improved with modifiers while others are not. Since the inclusion of modifiers can become unpredictable on a larger sample size, the decision has been made to keep the design without modifiers.

originalSentence	Output with modifiers	Output without modifiers
And walking? Do you walk without aid or do you have a walking stick or walker?	Patient walks stick without aid.	Patient walks stick without aid.
Completely without. And do you ever fall?	Patient completely fall.	Patient falls.
And how goes walking?	Walking goes.	Walking goes.
Can you climb stairs?	Patient climbs stairs.	Patient climbs stairs.
Do you have stairs in the house?	Patient has stairs in house.	Patient has stairs in house.
Yes. Stairwell. I actually never take the elevator.	Patient actually take elevator.	Patient takes elevator.
No, so you do that too and that also works?	Patient so do that.	Patient does that.
I write on a note.	Patient writes on note.	Patient writes on note.
Will it still go wrong?	It still go.	It goes.
One grocery goes, but two no longer.	Grocery no go.	Grocery goes.
One grocery as in one product?	Patient as product.	Patient as product.
Yes, but you do sometimes forget products?	Patient sometimes forget products.	Patient forgets products.
And do you buy any things double?	Patient buys things.	Patient buys things.
It's not that you have entire stocks of a certain product and buy even more fror	It even 's stocks of product.	It 'ss stocks of product.
So if he does that one thing, it is still going well, but then nothing else needs to l	Patient so go thing.	Patient goes thing.
That's not going well.	That well go.	That goes.
You do the housework, don't you?	Patient does housework.	Patient does housework.
And do you also have domestic help in addition or do you actually do everything	Patient also have help in addition.	Patient has help in addition.
If you have to pay in the store, how do you do it? With a pin or just cash?	Patient just do it in store.	Patient does it in store.
And do you pin yourself? In advance or how do you get cash?	Patient pins yourself in advance.	Patient pins yourself in advance.
Just from home.	Patient just from home.	Patient from home.
Yes, so you always pin or how am I supposed to picture that?	Patient so pin that.	Patient pins that.
Yes, and I sometimes have him pin while I'm there.	Patient sometimes have.	Patient has.
And now and then when he is alone he has to pin and then I first have to tell yo	Patient now have you.	Patient has you.
Because you forget the code, if you don't remember?	Patient forgets code.	Patient forgets code.
Well, then he's lost the code again.	Patient then lose code.	Patient loses code.
Then I lost the code.	Patient then lose code.	Patient loses code.
And who does the finances and administration?	Who does finances.	Who does finances.
Have you always done it or did you have to take over at some point?	Patient always do it over point.	Patient does it over point.
I always have.	Patient always have.	Patient has.
Score	14,5	15

Legend
Correct (+) 1 point
Partially correct (o): 0.5 point
Incorrect (-): 0 points

Figure 7.7: Comparison of output with and without modifiers.

7.4 Final version and results

Given my limited knowledge of linguistics, and the constraints of trying to transform language using rule-based systems, this is the final iteration of this thesis. The complete parser and realiser code can be found in Appendix 7.2 and 7.3 respectively. All twelve transcripts have been run through the parser and the sentence realiser. This iteration of the design cycle has further improved the output. A number of examples of improved generated narrative sentences are presented in Figure 7.8. The *iteration 4 output* column demonstrates the output of the previous iteration, while the final column shows the improved object and adposition handling.

Line	T	originalSentence	Iteration 4 output	Final output
5	1	Do you have stairs in the house?	Patient has house stair.	Patient has stairs in house.
8	1	I write on a note.	Patient writes note.	Patient writes on note.
35	11	Yeah move, I gotta move for the blood, blood.	Patient moves blood.	Patient moves for blood.
43	11	Yes Yes. And do you have help at home? Is there someone there for the hous	Patient has home help.	Patient has help at home.
70	2	Ah, nice. Okay, can you walk without an aid?	Patient walks aid.	Patient walks without aid.
76	2	I only take, I only took the elevator when I got home with the groceries.	Patient takes groceries elevator.	Patient takes elevator with groceries.

Figure 7.8: Improved sentences

Each sentence is again labelled by comparing the output to the preferred output. The same tags were used to determine whether a sentence is incorrect, partially correct or fully correct. Table 7.5 presents the results of the final design. 30% of the input sentences result in an excellent output sentence in the final design. Another 30% of the output is nearly correct but frequently misses some critical details. Finally, 40% of the results are still inaccurate. The final results are compared to the two previous assessments after iteration 1 and iteration 4 in Table 7.6. Here we see a gradual decrease in incorrect output, while the amount of correct and partially correct sentences have increased.

Table 7.5: final design results

Tag	Sentences	%
-	92	40%
o	71	30%
+	71	30%
Total	234	100

Table 7.6: relative comparison between iterations

Tag	It. 1	It. 4	Final
-	63%	45%	39%
o	23%	44%	30%
+	14%	11%	30%

There are several potential reasons for the high frequency of errors in the output. First of all, our knowledge of linguistics is limited, which led to a treatment design that relied on trial and error. While there was a general sense of which syntactic components are important for the narrative, programming it in a rule-based system proved difficult.

Second, a large quantity of the speaking turns consisted of either multiple sentences, or compound sentences. These proved particularly difficult to analyse and transform, due to the occurrence of tokens from multiple sentences being extracted, which resulted in the output sentence comprising a combination of the input sentences.

Third, the cleaned data still includes sentences that are difficult to transform due to their structure, grammatical errors or lack of context and referrals. Some sentences lack verbs, subjects or other parts-of-speech. People often do not speak in complete sentences during conversations. Additionally, certain words refer to earlier mentioned people, objects or statements. With the current implementation it is only possible to transform sentences individually without looking at context.

Four, because the parser and sentence realiser work only on individual sentences, it is not possible to include patient's explicit answers in the statements. Additionally, while a negate check was added to the parser, it could not be implemented successfully in the realiser. The issue with this approach is that the use of negating words such as "don't" and "not" in natural language is often more complex than simply negating an entire sentence. These words can be used as filler or for emphasis, rather than to negate the meaning of the entire sentence. As a result, using these words to negate the output of the realizer can lead to incorrect or misleading statements.

However, the successfully generated sentences do closely resemble the in Chapter 6 manually created narratives, which can effectively be used as input for the process of triplification.

8. Discussion and Conclusion

8.1 Conclusion

This research studied how a variety of linguistic techniques can be used to summarise geriatric conversations for the purpose of reducing administrative burden. The design cycle was used to structure this research, going through multiple iterations of problem investigation and analyses, treatment design and validation. Problem investigation was done for a large part through a literature study to find answers for RQ 1 and 2. For the following research questions, problem investigation was done by analysing available transcripts and material of a comprehensive geriatric assessment case study. Finally, the information gathered through these methods was utilized to design a treatment for the administrative burden in geriatrics. Conclusions for the 5 research questions and main research question will be stated below.

RQ1

What administrative issues do the nursing care and geriatrics sectors face?

Both scientific and grey literature proved that the administrative burden in healthcare in the Netherlands is in fact large. More than half of the professionals in home care and nursing homes express dissatisfaction with their administrative workload. Furthermore, IT systems that should serve as tools to make these tasks easier, are found to be difficult or inconvenient. Reports suggested that over 25% of professionals' time go to administration instead of quality care. Besides the discomfort for professionals, it costs the Netherlands €5 billion on a yearly basis. Statistical bureaus predict an increase in elderly people due to an aging population, which will pressure nursing professionals even further if no solutions can be found.

Reports however also show that the administrative issue is not unknown, and that institutions are working on solutions. Ideas to integrate technology more into the core tasks of professionals are becoming a standard for EMR competitors. EMR's are more often available on mobile devices. Additionally, research is being done on standardised reporting, allowing healthcare IT systems to better communicate with each other.

RQ2

What is the state-of-the-art in speech technology, NLP and AI within the healthcare domain?

Healthcare is a domain in which natural language, unstructured data, is common. Natural language processing techniques help in structuring this data, allowing for a wide range of opportunities. Due to the unstructured nature of this data, machine learning NLP tools are thriving. Speech technology is often still used as a tool for dictation, but has shown promise for applications such as enhancing speech of people with speech disorders, or for recognizing emotions or psychological conditions. The healthcare domain also involves highly standardized ways of providing care. General practitioners generally follow protocols for well-known diagnoses, but most healthcare professionals also work with guidelines. Ontology learning allows for (automatically) creating conceptual models of these guidelines. These ontologies can be utilized to structure medical conversations, allowing these to be interpreted by machines. One example of these ontologies is the Medical Guideline Ontology (MGO), which includes the human anatomy, symptoms, observations, diagnoses and treatments. The program Care2Report utilizes speech technology, MGOs and various NLP techniques to allow for the automatic transcription and interpretation of speech. The resulting summary is then uploaded to the EMR of the patient. Interpretation is done by transforming conversations into semantic triples, consisting of subjects, predicates and objects. These triples form a network known as a knowledge graph. By matching the knowledge graph to the MGO, the MGO becomes populated allowing for the generation of a report.

RQ3

How are geriatric conversations structured?

Geriatric conversations take place during comprehensive geriatric assessments (CGA). The Barthel-index (ADL) and Lawton-index (IADL) are utilized to determine the dependability of elderly people. These indices consist of multiple categories, which all test the capability of a patient regarding certain daily living activities. These tests are performed by doctors (geriatricians) in the form of interviews. Doctors ask the patient if they are still able to perform the activities, going through the categories one by one, and the patient answers the questions. Questions come generally in four types: Literal, paraphrased, suggestive, or confirmation. Patients answer either in an explicit way (yes or no), or implicit out of which the answer has to be derived. Two types of words or phrases can be identified that could help in the summarisation of geriatric dialogues: action indicators and dependency indicators. Action indicators are words that describe the category that is being discussed, while dependency indicators say something about how a patient is able to perform said activities.

RQ4

How can a consultation transcript be matched with geriatric ontologies?

Multiple conditions in a geriatric ontological conversation interpretation pipeline need to be satisfied in order to match conversations to a geriatric ontology. First of all, a Medical Guideline Ontology has to exist of the Barthel and Lawton indices. Second, triples have to be generated of a geriatric consultation transcription. This has proven to be difficult, due to information residing in multiple speaking turns. To counteract this problem, narratives can be extracted from conversations so that speaking turns are removed and only the most important statements remain. Six data cleansing steps, and 8 transformations were identified that allows the transformation of geriatric conversations to geriatric narratives. These in total 14 steps were demonstrated by including examples out of real-life geriatric dialogues. The narratives can be used to generate triples, which in turn can be matched to the geriatric ontology.

RQ5

How to automatically extract narratives from geriatric dialogues?

Geriatric dialogues can be summarized by first creating narratives. The linguistic rule-based approach investigated in this thesis was to parse sentences and find the most essential syntactic components. By extracting these components, irrelevant language such as interjections are left out. The tool investigated for this task was SpaCy's dependency parse and part-of-speech tagger. By tagging the various tokens using the SpaCy library, and extracting them using a Python script it was possible to specify which preferred syntactic components could be used for the narrative. The relevant syntactic components can then be inserted into a sentence realiser, to create short and meaningful statements about patients. These statements can be efficiently used to create knowledge triples. The sentence realiser used in our design was a C# wrapper of the SimpleNLG library. In the final version subjects, objects and predicates could be extracted with decent accuracy. Additionally, it was possible to include modifiers and adpositions in the realised sentences with varying degrees of success. Around 30% of the input sentences yielded a correct narrative sentence, and a further 30% yielded partially correct narrative sentences.

MRQ

Which linguistic techniques can be used in a pipeline as a solution to the administrative burden in geriatric performance assessment and nursing?

To answer the main research question, first a literature study was done to investigate the problem, and find the state-of-the art in linguistic techniques as a possible solution to the problem. The literature study found two main approaches to the automatic summarisation of geriatric conversations: machine learning and rule based. While machine-learning in the long term shows the most promise, this thesis showed multiple other linguistic techniques (both ML and rule-based) that can be used to summarise and interpret geriatric conversations.

Specifically, this thesis examined the use of ontologies and knowledge graphs to interpret geriatric conversations and thereby allowing for automatic report generation. The existing pipeline however had to be expanded with an additional step, thereby requiring more linguistic techniques to be utilized.

First, a sequence of manual steps have been developed, that proved to be successful in extracting the important semantics out of a conversation in the form of a narrative. To automate this process, a dependency parser was used. This common linguistic technique combines the rule-based nature of syntax with an English language model. The second linguistic technique used was a sentence realiser called SimpleNLG. This is a rule-based software program which takes various syntactic components and uses the rules of English language to generate new sentences. Both the parser and sentence realiser proved to be useful techniques to identify important syntactic elements, while rejecting irrelevant phrases. It also proved to have potential to generate short narratives out of individual sentences. However, this approach failed in recognizing conversational context, positive or negative answers and is in its current design therefore incomplete as a solution.
In short, the linguistic techniques that can be used in a pipeline as a solution to the administrative burden are ontology learning, triple generation, triples matching, narrative information extraction, sentence parsing and sentence realisers, as well as various machine-learning approaches.

8.2 Discussion

In this final chapter, we present and elaborate on the main contributions of our research. Additionally, the limitations of the research are highlighted, as well as recommendations for future studies are provided.

8.2.1 Main contributions

This study made several notable contributions to the field of linguistic techniques as an answer to the administrative burden in geriatrics, and healthcare as a whole. First of all, this study provides a comprehensive overview of the administrative burden within the nursing profession in the Netherlands. Despite the prevalence of this issue in grey literature reports, it has been underrepresented in the scientific literature.

Second, building on the research of Kendall Kemper and Rick Oostveen, we analysed geriatric transcripts and identified common indicator words that can help with the interpretation and summarisation of geriatric conversations. Additionally, this study discussed various factors that can impact doctor's decision-making during patient assessments.

Third, this study found that triple-generation for conversations is made difficult due to the presence of multiple people and speaking turns, and proposed a solution as to work around this. The first main contribution from a design-standpoint is the development of a novel narrative information extraction technique for geriatric assessments. When following these sequence of steps, it is possible to extract the main narrative out of transcripts from the specific anamnesis from a comprehensive geriatric assessment. Though this technique was developed with geriatric assessments in mind, the general nature of the steps renders it potentially useful for structured doctor-patient interviews of other healthcare domains.

Finally, this research contributed to the field of narrative information extraction by providing a possible pipeline allowing conversations to be transformed to narratives using a dependency parser and a sentence realiser. Even though in it's current form it is inadequate as a solution, building upon the current ruleset, or enhancing it with machine-learning could be a promising way of processing conversations in such a way that triplification becomes possible.

8.2.2 Validity Threats

To assess the validity and trustworthiness of the results, the four aspects of validity are discussed: construct validity, internal validity, external validity and reliability (Wohlin et al., 2012). Both threats to these aspects are discussed, as well as strategies that can help tackle these threats.

Construct validity

Construct validity reflects the extent to which the operational measures employed accurately represent the researcher's goals and research questions. A step-by-step explanation was given of how the methodology and design evolved, based on information retrieved from literature, or insights gained through different sources. There is a threat based on mono-operation bias. Only one treatment pipeline was tested, and although alternatives were considered, this might give an incomplete idea of the theory. Furthermore, there may be bias as to what is considered narrative, and what is considered 'relevant' information. While we did interview an expert in the area of geriatrics, the specific information that should be included in medical reports was unknown during the creation of the narrative information extraction technique.

Internal validity

This aspect reflects the level to which causal relationships are not affected by different factors. First of all, the different social factors and other factors relating to the subjects could not be controlled, because the data was already collected prior to the start of this research. However, there are some factors that could have influenced the results. The twelve interviews were conducted by three different doctors. These doctors may have different interviewing habits or tactics, which can lead for example to different answers and language use. During the treatment design and validation, no distinction was made between these groups, and all patients were evaluated together.

External validity

External validity aims to find whether it is possible to generalize findings and the degree to which the results are relevant to other people outside of this specific case. There is a threat to the interaction of setting and treatment. This threat is in effect when the experimental setting or material is not up to industry standards. While we put great effort in using modern linguistic tools to transform and generate textual data, our knowledge is limited on all available online tools. Additionally, due to our inexperience with linguistics and machine-learning, a rule-based algorithm was preferred for this case study. Different researchers however might have more success using machine-learning methods, or are perhaps aware of tools we knew not existed.

Furthermore, there may be a threat regarding the subject population. The twelve transcripts were collected from a single polyclinic. It could be possible that the subjects in that area speak a dialect or have certain linguistic tendencies that other parts in the Netherlands or the world do not have. Combining this with the fact that twelve transcripts are not an ideal number, the results may not be fully generalisable. However, this problem was hard to tackle, as data collection was performed before this research started, as well as the general difficulty of setting up experiments in healthcare.

Reliability

This aspect reflects the level to what extent the researcher influences the data and analysis. Experiments should be repeatable, meaning that methodology and data collection should be clear. We have tried to accomplish this by explaining each step in great detail. Choices as to why certain decisions were made, and how results were achieved were fully disclosed. Reliability of treatment implementation was done by including code-snippets for each addition or change to the code, as well as including the full code in the appendix. A possible threat here is the use of a C# wrapper and swaggerUI to test the code. The full code of this framework was not disclosed in the thesis, as it should not affect the results. Another threat is the way we assessed the output. The scale of incomplete, partially complete and fully complete contain an element of subjectivity, as the weighting of errors may vary among different researchers. Furthermore, the steps of the narrative information extraction can be open for interpretation. For example the data cleansing step: Removed semantically incomprehensible or unclear phrases. Different researchers may find different phrases to be unclear. Due to the research material consisting of premade transcripts, there is no threat to the experimental setting or subjects. Interviewing and describing a different set of patients however could change the results.

8.2.2 Future work

This study examined a first approach of automating geriatric assessments by attempting to automatically extract narratives from conversations and use those to generate knowledge triples. While various approaches were considered, only one could be explored in greater depth. In this final section we will identify possible directions future research could go.

Involve linguistic experts to improve the system

Our algorithm was based on the most elemental form of sentence composition, and build from there using primarily trial and error and output analysis. To improve the accuracy of automated narrative information extraction, linguistic experts should be involved in the process. First of all, their knowledge of existing tools is greater and they could therefore advice on which libraries and packages to use to optimise results. Additionally, their knowledge of language could help in setting up rules to which the sentence realiser should adhere in order to create grammatically and semantically correct sentences.

Try a machine-learning approach

A rule-based approach was chosen as a result of multiple factors, including inexperience with machine-learning and a small dataset. Language consists of rules, so it shouldn't necessarily be impossible to create a functional system, however certain problems need to be overcome. First of all, our implementation proved to be unsuccessful in including context and semantics into the narrative generation. Second, the large amount of different sentence structures, exceptions and language errors result in the demand of a large quantity of rules. These problems could be solved by training a model on a large training set. Our research included only 12 transcripts, which would be insufficient for training, but perhaps the model can be trained on other sources as well. Although during research no satisfactory pretrained models were found, perhaps NLP experts could advise on this matter. A machine-learning approach could simply learn to combine both questions and answers into a single statement.

Find alternatives (improve triplification software)

Instead of creating narratives to allow for better triplification, an alternative method can be found that either skips this step entirely, or changes it. Our approach was a linguistic one, tackling it primarily from a syntactic point of view and removing, however there is no prove that this is the most efficient way. Furthermore, rather than finding ways to create narratives, research can be done on improving triplification software triples can be created from conversations directly.

Improve transcript pre-processing

One of the main challenges in this thesis was the nature of conversations. Compound sentences, filler words and grammatical errors made the parsing and realising of sentences difficult. Data-cleansing for the final pipeline was done manually. Multiple times during research, sentences were found that should not have been removed earlier in the process. Scripts could help automate this process. For example using libraries that detect filler words and interjections, or libraries that can transform compound sentences into complete separate sentences.

8.3 Acknowledgements

First, I would like to express my sincere thanks to my first supervisor, Prof. dr. Sjaak Brinkkemper, for his time, feedback and guidance throughout this project. I would additionally like to express my gratitude for his patience and understanding during any difficulties that arose during this project.

Second, I would like to thank my second supervisor Dr. Fabiano Dalpiaz for providing feedback on the project on multiple occasions.

Then I would like to thank my Mom, Dad and brothers, for supporting me throughout my thesis and my entire journey to my master’s degree. Finally I’d like to thank my friends, specifically Anh, Renate and Denise, as well as my college friends for providing me with all their support, advice and motivation.

9. References

Ali, A., & Renals, S. (2018, July). Word error rate estimation for speech recognition: e-WER. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Volume 2: Short papers*, 20-24.

Angeli, G., Premkumar, M. J., & Manning, C. D. (2015, July). Leveraging Linguistic Structure For Open Domain Information Extraction. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 1: Long Papers*, 344-354.

Asim, M. N., Wasim, M., Khan, M. U., Mahmood, W., & Abbasi, H. M. (2018). A survey of ontology learning techniques and applications. *Database, 2018*.

Carlini, N., & Wagner, D. (2018). Audio Adversarial Examples: Targeted Attacks on Speech-to-Text. *IEEE Security and Privacy Workshops*, pp. 1-7.

CBS. (2020, December 16). *Kerncijfers van diverse bevolkingsprognoses en waarneming*. Retrieved from CBS StatLine: https://opendata.cbs.nl/statline/#/CBS/nl/dataset/70737ned/table?dl=56917

CBS. (2021, November 18). *Werknemers met een baan in de zorg en welzijn (Employees with a job in care)*. Retrieved from CBS StatLine: https://azwstatline.cbs.nl/?dl=470D9#/AZW/nl/dataset/24016NED/table

Centers for Medicare & Medicaid Services. (2012, March 26). *Electronic health records.* Retrieved September 9, 2021, from cms.gov: https://www.cms.gov/medicare/e-health/ehealthrecords?redirect=/ehealthrecords/

Eisenstein, J. (2019). *Introduction to natural language processing.* MIT press.

ElAssy, O., Vendt, R. d., Dalpiaz, F., & Brinkkemper, S. (2022). A Semi-automated Method for Domain-Specific Ontology Creation from Medical Guidelines. *International Conference on Business Process Modeling, Development and Support, International Conference on Evaluation and Modeling Methods for Systems Analysis and Development*, 295-309. Retrieved May 5, 2022

Engelen, E. v., & Bokhorst, M. (2016, Maart). *Uitgebreide toelichting van het meetinstrument - Barthel Index.* Retrieved December 19, 2022, from meetinstrumentenzorg.nl: https://meetinstrumentenzorg.nl/wp-content/uploads/instrumenten/Barthel-index-form.pdf

Eurlings, F., & Schaik, M. v. (2020, October 28). *Vijf jaar ECD-marktonderzoek: stijgers, verschuivingen en nieuwe ontwikkelingen (Five Year EHR-market research: Risers, shifts and new developments*. Retrieved from M&I: https://mxi.nl/kennis/472/vijf-jaar-ecd-marktonderzoek-stijgers-verschuivingen-en-nieuwe-ontwikkelingen

Federatie Medisch Specialisten. (2021, September 1). *Comprehensive geriatric assessment (CGA).* Retrieved January 1, 2022, from Richtlijnendatabase: https://richtlijnendatabase.nl/richtlijn/comprehensive_geriatric_assessment_cga/startpagina_-_comprehensive_geriatric_assessment_cga.html

Filippidou, F., & Moussiades, L. (2020, June). A Benchmarking of IBM, Google and Wit Automatic Speech Recognition Systems. *IFIP International Conference on Artificial Intelligence Applications and Innovations*, 73-82.

Gatt, A., & Reiter, E. (2009, March). SimpleNLG: A realisation engine for practical applications. *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, 90-93.

Google. (2022, August 18). *Speech-to-Text basics*. Retrieved August 23, 2022, from Google Cloud: https://cloud.google.com/speech-to-text/docs/basics?hl=nl#select-model

Groot, K. d., Paans, W., Veer, A., & Francke, A. d. (2017). *Knelpunten bij verslaglegging door verpleegkundigen en verzorgenden.* Nivel.

Gruber, T. R. (1995). Toward principles for the design of ontologies used for knowledge sharing? *International journal of human-computer studies, 43(5-6)*, 907-928.

Halle, M. (1962). Phonology in Generative Grammar. *WORD*, 54-72.

Hanek, Hanekamp, M., Heesbeen, S., Helm, I. v., & Valks, R. (2019). *Administratieve belasting langdurige zorg 2019 (Administrative burden long-term care 2019).* Berenschot.

Hanekamp, M., & (Berenschot). (2018). *Enquête administratieve belasting langdurige zorg.* Retrieved from Zorg voor Beter: https://www.zorgvoorbeter.nl/zorgvoorbeter/media/documents/thema/verbeteren-doe-je-zo/enquete-administratieve-belasting-langdurige-zorg.pdf

Hardiker, N. R., Dowding, D., Dykes, P. C., & Sermeus, W. (2019). Reinterpreting the nursing record for an electronic context. *International journal of medical informatics*.

Huang, X., Acero, A., & Son, H.-W. (2001). *Spoken Language Processing: a guide to theory, algorithm and system development.* Prentice Hall.

Jones, K. S. (2001). *Natural language processing: a historical review.* University of Cambridge, Computer laboratory. Cambridge: Artificial Intelligence review.

Kemper, K., Brinkkemper, S., & Dalpiaz, F. (2021). *Filtering Irrelevant Information in Medical Consultations.* University Utrecht, Utrecht.

Kimura, T., Nose, T., Hirooka, S., Chiba, Y., & Ito, A. (2018, November). Comparison of Speech Recognition Performance Between Kaldi and Google Cloud Speech API. *International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 109-115.

Koenecke, A., Nam, A., Lake, E., Nudell, J., Quartey, M., Mengesha, Z., . . . Goel, S. (2020). Racial disparities in automated speech recognition. *Proceedings of the National Academy of Sciences, 117*, 7684-7689.

KPMG. (2020). *Report: Ontregel de zorg (Unburden the health sector).* KPMG International.

Latif, S., Qadir, J., Qayyum, A., Usama, M., & Younis, S. (2020). Speech Technology for Healthcare: Opportunities, Challenges, and State of the Art. *IEEE Reviews in Biomedical Engineering*, 342-356.

Lawton, M. P., & Brody, E. M. (1969). Assessment of Older People: self-maintaining and instrumental activities of daily living. *The gerontologist*, 179-186.

Liddy, E. D. (2001). *Natural language processing.* Syracuse University, School of Information Studies.

M, A., Orrell, M., Tolson, D., Abbatecola, A., Arai, H., Bauer, J., . . . Vellas, B. (2015). An International Definition for “Nursing Home”. *Journal of the American Medical Directors Association*.

Maas, L., Geurtsen, M., Nouwt, F., Schouten, S., Water, R. v., Duimen, S. v., . . . Brinkkemper, S. (2020). The Care2Report system: automated medical reporting as an integrated solution to reduce administrative burden in healthcare.

Maas, L., Kisjes, A., Hashemi, I., Heijmans, F., Dalpiaz, F., Dulmen, S. v., & Brinkkemper, S. (2021). Automated Medical Reporting: From Multimodal Inputs to Medical. *HEALTHINF*, 509-514.

Mahoney, F. I., & Barthel, D. W. (1965). Barthel index. *State Medical Journal*.

Matarneh, R., Maksymova, S., Lyashenko, V. V., & Belova, N. V. (2017). Speech Recognition Systems: A Comparative Review. *OSR Journal of Computer Engineering (IOSR-JCE), 19*(5), PP 71-79.

Mohit, B. (2014). Named entity recognition. In B. Mohit, *Natural Language Processing of Semitic Languages* (pp. 221-247). Berlin: Springer.

Navigli, R., & Velardi, P. (2002, August). Semantic Interpretation of Terminological Strings. *Proc. 6th Int’l Conf. Terminology and Knowledge Eng*, 95-100.

Nictiz. (2021a). *Wat is een zib?* Retrieved 10 02, 2021, from Nictiz: https://www.nictiz.nl/standaardisatie/zib-centrum/wat-is-een-zib/

Nictiz. (2021b, May 12). *Opbouw eOverdracht v4.0 - aanmelding*. Retrieved from Informatiestandaarden Nictiz (Data standards): https://informatiestandaarden.nictiz.nl/wiki/vpk:V4.0_Opbouw_eOverdracht_aanmeldbericht

Oostveen, R., Brinkkemper, S., & Dalpiaz, F. (2022). *Care2Report: Structuring and analyzing medical consultation sessions for Automated Medical Reporting.* Utrecht: Utrecht University.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., . . . Vesely, K. (2011). The Kaldi Speech Recognition Toolkit. *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*.

Saz, O., Yin, S.-C., Lleida, E., Rose, R., Vaquero, C., & Rodríguez, W. R. (2009). Tools and technologies for computer-aided speech and language therapy. *Speech Communication*, 948-967.

Smith, M. C. (2019). *Nursing theories and nursing practice* (Fifth ed.). F.A. Davis.

SNOMED CT. (2022). *SNOMED International*. Retrieved July 1, 2022, from The value of SNOMED CT: https://www.snomed.org/snomed-ct/why-snomed-ct

Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of buisness research*, 333-339.

Stanovsky, G., & Dagan, I. (2016, November). Creating a Large Benchmark for Open Information Extraction. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2300-2305.

Topaz, M., Schaffer, A., Lai, K. H., Korach, Z. T., Einbinder, J., & Zhou, L. (2018). Medical Malpractice Trends: Errors in Automated Speech Recognition. *Journal of medical systems, 42*, 1-4.

Veer, A. d., Brinkman, K. d., & Francke, A. (2017). *Administratieve druk: méér dan kwestie van tijd (Administrative burden: More than a matter of time.* Nivel.

Vogel, M., Kaisers, W., Wassmuth, R., & Mayatepek, E. (2015). Analysis of Documentation Speed Using Web-Based Medical Speech Recognition Technology: Randomized Controlled Trial. *Journal of medical Internet research*.

Werner, L., Huang, G., & Pitts, B. (2019, November). Automated speech recognition systems and older adults: a literature review and synthesis. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting, 63*(No. 1), 42-46.

Wieringa, R. J. (2014). *Design Science Methodology for Information Systems and Software Engineering.* London: Springer.

Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, 1-10.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2012). *Experimentation in Software Engeneering.* Springer Science & Business media.

Yu, D., & Deng, L. (2016). *Automatic Speech Recognition.* Springer london limited.

Zibs. (2020). *Patient-v3.2(2020NL)*. Retrieved from zibs.nl: https://zibs.nl/wiki/Patient-v3.2(2020NL)

Zorgrating. (n.d.). *Het zorglandschap in cijfers (Healthcare in numbers)*. Retrieved November 30, 2021, from https://zorgrating.nl/het-zorglandschap-in-cijfers/

10. Appendices

10.1: Appendix 7.1: Unique POS and Dependency finder (Python)

```
import os
import json
import spacy
# path to all transcripts
my_path = "Path to folder with txt files"
nlp = spacy.load("en_core_web_trf")
file_list = []
pos_wordtypes = []
dep_wordtypes = []
for file in os.listdir(my_path):
    filename = os.fsdecode(file)
    if filename.endswith(".json"):
         file_list.append(filename)
         continue
    else:
         continue

def unique_pos(text):
    for sentence in text:
        doc = nlp(sentence)
        for words in doc:
            pos_wordtype = words.pos_
            if pos_wordtype not in pos_wordtypes:
                pos_wordtypes.append(pos_wordtype)
    return "POS: " + str(pos_wordtypes)


def unique_dep(text):
    for sentence in text:
        doc = nlp(sentence)
        for words in doc:
            dep_wordtype = words.dep_
            if dep_wordtype not in dep_wordtypes:
                dep_wordtypes.append(dep_wordtype)
    return "DEP: " + str(dep_wordtypes)

for file in file_list:
    with open(my_path + file) as jsonfile:
        text = json.load(jsonfile)
        print("File: ", file, unique_pos(text))

for file in file_list:
    with open(my_path + file) as jsonfile:
        text = json.load(jsonfile)
        print("File: ", file, unique_dep(text))
```

10.2: Appendix 7.2: Dependency parser code (Python)

```
import spacy
import json
from spacy import displacy
from pathlib import Path

nlp = spacy.load("en_core_web_trf")
transcriptie = open("./cleansedP1.json")
text = json.load(transcriptie)

output = []
id = 0
for sentence in text["Transcriptie"]:
    print(sentence)
    doc = nlp(sentence)
    id = id + 1
    nsubj = []
    pron = []
    root = []
    pred = []
    verb = []
    dobj = []
    pobj = []
    adverb = []
    adjective = []
    adposition = []
    non_personal_pronouns = ["this", "that", "these", "those", "it", "they", "who", "what",
"which"]
    negated = False

    for token in doc:
        if token.dep_ == "neg":
            negated = True
        # SET SUBJECT
        # If the word is a subject and not a pronoun. Word == subject
        if (token.dep_ == "nsubj" or token.dep_ == "nsubjpass")  and token.pos_ != "PRON":
            nsubj.append(token.text.lower())
        # If the word is a subject and a personal pronoun. Subject == patient
        if(token.dep_ == "nsubj" or token.dep_ == "nsubjpass") and token.pos_ == "PRON" and
token.text.lower() not in non_personal_pronouns:
            nsubj.append("patient")
        # If the token is a subject and pronun, but NOT a personal pronoun. Word == subject
        if(token.dep_ == "nsubj" or token.dep_ == "nsubjpass") and token.text.lower() in
non_personal_pronouns:
            nsubj.append(token.text.lower())

        # SET ROOT
        # If statement to find and set root if root is not a predicate
        if token.dep_ == "ROOT" and (token.pos_ != "AUX" and token.pos_ != "VERB"):
            root.append(token.text.lower())
        # If statement to find and set predicate
        if token.dep_ == "ROOT" and (token.pos_ == "VERB" or token.pos_ == "AUX"):
            pred.append(token.text.lower())

        # SET VERBS
        if  token.pos_ == "VERB":
            verb.append(token.text.lower())

        # SET OBJECTS
```

```
        # SET OBJECTS
        # If statement to find and set objects
        if token.dep_ == "dobj":
            dobj.append(token.text.lower())
        if token.dep_ == "pobj":
            pobj.append(token.text.lower())

        # SET ADVERB
        if token.pos_ == "ADV":
            adverb.append(token.text.lower())
        # SET ADJECTIVE
        if token.pos_ == "ADJ":
            adjective.append(token.text.lower())
        # SET ADPOSITION
        if token.pos_ == "ADP":
            adposition.append(token.text.lower())

    # If there is no subject, default is "patient"
    if not nsubj:
        nsubj.append("patient")
    if not pred and len(verb) > 0:
        pred.append(verb)

    output.append({"id" : id, "sentence" :sentence, "nsubj" :nsubj, "pron" :pron, "root"
:root, "pred" :pred, "verb" :verb, "dobj" :dobj, "pobj" :pobj, "adverb" :adverb, "adjec-
tive" :adjective, "adposition" :adposition, "negated": negated})

json_string = json.dumps(output)

with open('./ParseP1.json', 'w') as outfile:
    outfile.write(json_string)
```

10.3: Appendix 7.3: SimpleNLG sentence realiser code (C#)

```
using AutoMapper;
using Core;
using SimpleNLG;

namespace Repository.SimpleNLG;

public class SimpleNlgLibrary : ILanguageLibrary
{
    private readonly ISimpleNlgRepository _simpleNlgRepository;
    private readonly IMapper _mapper;
    public SimpleNlgLibrary(ISimpleNlgRepository simpleNlgRepository, IMapper
mapper)
    {
        _simpleNlgRepository = simpleNlgRepository;
        _mapper = mapper;
    }

    public string CreateSentence(SentenceConfiguration config,
FeatureConfiguration featureConfig)
    {
        var nlgFactory = _simpleNlgRepository.GetNlgFactory();
        if (config.IsEmpty()) return string.Empty;
         var sPhraseSpec = nlgFactory.createClause();
        var objects = nlgFactory.createCoordinatedPhrase();

        if (config.Subject != null) sPhraseSpec.setSubject(config.Subject);
        if (config.Predicate != null) sPhraseSpec.setVerb(config.Predicate);
        if (config.DirectObject != null)
objects.addCoordinate(config.DirectObject);
        if (config.PObject != null)
        {
            var PObjectPP = nlgFactory.createPrepositionPhrase();
            PObjectPP.addComplement(config.PObject);
            if (config.Adposition != null)
PObjectPP.setPreposition(config.Adposition);
            objects.addCoordinate(PObjectPP);
        }
        sPhraseSpec.setObject(objects);
        objects.setFeature("CONJUNCTION", null);

        foreach (var feature in _mapper.Map<List<KeyValuePair<string,
object>>>(featureConfig))
        {
            sPhraseSpec.setFeature(feature.Key, feature.Value);
        }

        return _simpleNlgRepository.GetRealiser().realiseSentence(sPhraseSpec);
    }
}
```