Master of Science thesis

The adaptive presentation assistant

Using grammar-based recognition to support the process of presenting

Laurens Satink

August 21th, 2009

Graduation committee

Dr. R.J.F. Ordelman (first supervisor)

Dr. A.J. van Hessen

Prof. Dr. Ir. A. Nijholt

Abstract

Giving presentations has changed significantly over the last decades. Although the main reason why someone wants to give a presentation remains the same, the way it is done has changed significantly. During the last 30 years of the twentieth century, overhead slides displayed on an overhead projector were used for broadcasting the information. Over the past decades, these analogue overhead slides have been replaced by digital slides, projected by a beamer on a screen. More than the way of projecting, it is the slide itself that has changed teh most. Transparent overhead slides limited the information displayed to text, drawings, and with the possibility to photo copy on slides, figures and photo's. Using digital slides made it possible to add enriched multimedia content, such as sound fragments, animations, (moving) pictures, movies, and appearing and vanishing text.

These changes strongly influence the way a presentation can be given. In this thesis, we explore the possibility to apply speech and language technology (SLT) such as automated speech recognition (ASR) to allow the presenter to use (natural) speech to navigate through presentation.

Several approaches were explored, developed and evaluated. First, a command-driven presentation assistant is created, after which titles and content of other slides are included. For each approach, a series of experiments is conducted and interpreted to evaluate the performance.

The outcome shows that the presentation assistants in most cases are able to support the presenter in showing the correct slide at almost the correct time. A small but noticeable delay is inevitable when speech is used as a basis for the determination process, as the presenter discusses the next slide before the transition is initiated.

The accuracy of the assistants is on average around 80-90% during conversational speech.

Preface

This thesis contains the research, development, implementation and evaluation of several presentation assistants that together define my graduation project for the HMI chair within the faculty of Electrical Engineering, Mathematics and Computer Science at the University of Twente.

The start of this project was back in July 2005. Due to unforeseen circumstances, completing it has taken significantly more time than expected or planned. For standing patiently besides me, offering me help and advice whenever needed or possible, I would like to sincerely and profoundly thank my graduation committee, especially Roeland and Arjan, for their time and effort. These last months have been intense, and their support has been invaluable.

In addition, I want to extend my thanks to professor Wambacq, of the University of Leuven, for offering support and help using SPRAAK, and for providing the necessary models.

Also, I would like to thank a lot of faculty members for enabling me to graduate in this way: Sharon Vonk, Hans Romkema, Gerda Olthof and all the others that helped me to finish my study.

A special thanks to my parents, for enabling me to study at this university, and for their continuous support throughout all these years. I could not have come this far without you.

Off course I cannot forget my friends and fellow students, for making my study time so worthwhile, just as much within the educational setting as during all the spare time we spent together.

My colleagues from Telecats, for allowing me to continue and finish my study next to my job, and for allowing me to take days and hours off whenever it was needed.

Last, I want to thank Tabitha for sticking with me and aiding me where possible during these last stressful months.

Laurens

Contents

1 Introduction

Giving presentations is a common good. It has been incorporated into most aspects of corporate, as well as educational, academic and even personal life. Most presenters support their presentations with digital slides, prepared in advance. During the presentation, the exact sequence in which the slides are displayed is specified directly using a keyboard, mouse or other device. The presenter uses these devices to output a signal, which is interpreted by the software to execute an action, typically displaying the next slide. However, this is a static action-response environment.

What if this direct interaction could become obsolete?

1.1 Problem description

Our environment has been made more and more aware of our presence during the last few decades. Connecting the light outside ones house to a motion detector, equipping a car with a transmitter and the garage door with a corresponding receiver and sliding doors are just basic examples of this notion. However, far more advanced issues such as engaging in dialogues with computers or computer systems, interactive route planning and all sorts of sensor systems are present in everyday life as well. The domain of giving presentations is no exception to this phenomenon.

During a typical presentation, presenters direct the sequence in which the slides are displayed by executing actions. Back in the days of the analogue overhead projector, these actions consisted of selecting the next slide from an (ordered) set of slides and putting it on the overhead projector. Since the dawning of the digital era and the arrival of software solutions such as PowerPoint, this set of actions has been replaced by selecting the next slide from the ordered set of digital slides. This action is easily carried out by pressing a button on a device connected in some way to the computer containing the presentation, after which the desired slide is shown. It does however still require the presenter to manually carry out that action.

This thesis attempts to rule out that necessity by adapting the sequence of the presentation to the speech of the presenter. This is carried out by applying methods and techniques from the field of Speech and Language Processing (SLP) technology to the domain of presenting. Issuing commands will be made possible with the aid of keyword spotting (KWS). Further analysis of the content of slides will be a means to construct grammars, which will be loaded into an automated speech recognition (ASR) system.

1.2 Towards understanding a presenter

Imagine a presentation environment in which the presenter is no longer required to prepare a presentation, but can just bring along a repository of rich content that illustrates his topic. When the presentation starts, he loads his assistant and just starts speaking. The assistant listens and understands what the presenter is talking about, and displays appropriate illustrative content.

Not only is the assistant able to display content relevant to the presenter's speech, it has also learned some characteristics of the presenters. It knows that on average, a slide is displayed between 30 seconds and 3 minutes, but never shorter than 20 seconds. This heuristic knowledge is applied when deciding when to switch content.

Figure 1 A presentation assistant listening to a presenter and displaying relevant content

Although this presentation assistant is far from being realised, some projects have contributed towards it. For example, there is the Radio Oranje demo[1], a spoken document retrieval system based on the recordings of Queen Wilhelmina during the 2nd World War. The recordings have been automatically indexed offline based on the research of Ordelman et al (2005) and allow the user to query these recordings. The relevant parts of recordings can be selected and played. The speech is aligned with the audio, highlighting the uttered words. During the playback, a photograph or other image related to the current segment is displayed. The relation between the speech and selected image is largely based on available metadata of the images.

Another project within the HMI chair demonstrates the use of advanced multimedia retrieval techniques to make the Dutch news broadcasts searchable. Each broadcast is segmented based on topic detection or speaker changes, after which recognising and indexing is performed. The user can query the recent broadcasts on either the subtitles (available for every broadcast) or the recognition results. The results are ranked and presented to the user, who is then able to view the parts of the broadcasts that were returned[2].

The technology used in both showcases is of great value to the future assistant, as they can be applied to a presenter's speech to retrieve a relevant video stream, picture or any other multimedia element to illustrate the current topic.

This thesis will attempt to contribute towards developing the futuristic presentation assistant focussing on presentations and the presenter's speech.

1.3 Problem statement

Most presentations are given nowadays with the aid of PowerPoint or a similar software product. Prior to the presentation, the presenter prepared a set of slides, which is most likely shown in a linear fashion. Displaying a next slide is initiated by the presenter by means of clicking a mouse (or a similar action). It would be more convenient if the presentation would adapt to what the speaker is saying: based on the speech of a presenter, the appropriate slide should be displayed automatically. This is not necessarily the next slide in the presentation.

This thesis assesses the (im) possibilities of the appliance of speech and language processing (SLP) methods and information retrieval (IR) techniques in order to this.

[1] The Radio Oranje demo is available at http://hmi.ewi.utwente.nl/showcase/Radio Oranje demo

[2] A demonstration is available at http://hmi.ewi.utwente.nl/showcases/Broadcast-news-demo

1.3.1 Initiating slide transitions with ASR

Given a traditional (linear) presentation, the next appropriate slide to display after the current one is almost every time the next slide in the presentation. If we assume this to be true, transitioning to the next slide could be triggered automatically. The speech of a presenter is transformed into text with the aid of Automated Speech Recognition (ASR). This (recognized) text can be analyzed by applying several techniques in order to wait a little bit longer, or to decide to display the next appropriate slide. The following methods will be evaluated for their performance:

- The ASR can be configured to identify commands such as 'next slide' or 'show previous', which will trigger the transition to the desired slide. The process of extracting keywords in the speech is called keyword spotting (KWS).
- An ASR can process complex structures conveyed in grammars to extend the parsing of keywords into parsing words in a context. The required grammars are constructed from the content of the slides and are used to determine when the presenter starts speaking about the next slide, thus initiating a slide transition.

1.4 Research goal

To determine when a slide transition is in order, several presentation assistants will be designed and implemented. These assistants analyze the (recognized) speech from the presenter and compare them to the slides of the presentation. The ASR models will be increasingly complex and are based on knowledge confined within the presentations.

This thesis attempts to prove the following hypothesis:

A presentation assistant using ASR is able to determine when a slide transition is in order.

In order to prove this hypothesis, each of the assistants will be evaluated for their accuracy and performance.

1.5 Outline

1. **Introduction** This chapter describes the context in which thesis can be placed and derives the research goals.
2. **Related work**. Related projects and research are briefly touched in this chapter.
3. **Presentations** In this chapter presentations and the process of presenting are examined more thoroughly.
4. **Automated speech recognition** This chapter covers the required methods and techniques from the field of SLP to construct the required configuration for the ASR.
5. **Approach** This chapter covers the different approaches that are used to construct the presentation assistant. It describes the detailed construction of these grammars, along with a global design of the assistants.
6. **Evaluation** This chapter first defines a set of performance measures used to evaluate the assistants. Next, it describes the chosen evaluation approach after which the performance measures are established for the different assistants.
7. **Conclusions** This chapter formulates an answer to the research goal, along with noticeable side observations. Last, some suggestions for further research on presentation assistants is provided.

2 Related work

This chapter glances at the context in which this project can be placed. It provides an overview of related projects and offers a view on the possible applications

2.1 Jabberwocky

One of the closest related projects to this thesis is the Jabberwocky system designed and implemented by Franklin et al. (2000). It is part of the greater Intelligent Classroom project, which aims at designing a multi-modal interactive classroom. The focus is to provide an intelligent environment in which "*the user expresses what he wishes to do, and the environment recognizes his intentions and attempts to accommodate the user*" (Infolab, 2007).

Jabberwocky is a speech-based interface to Microsoft's presentation software tool, PowerPoint (Microsoft Corporation, 2007). In essence, it is a system that learns how to aid speakers during the rehearsal of their presentations. It parses the content of the slides of the presentation for extraction of keywords and phrases and associates them with their places in the presentation. The system generates synonymous sentences for each content element of the presentation for matching the speaker's utterances with the elements. Syntactic and semantic rules are applied for the generation of alternatives, thus creating a large set of keywords and (partial) key phrases. Other NLP methods such as stemming are applied as well, in order to assist the user in using natural language instead of the condensed word groups often find in presentations. These sets may expand during each rehearsal, making the system in principle quite extensive and speaker-specific. Franklin et al. (2000) found that before rehearsals, Jabberwocky extracted 192 words and phrases. After three rehearsals, on the used presentation in the experiment, Jabberwocky increased its knowledge with 85 new phrases, and rediscovered a total of 37 phrases in the second and third rehearsal from the previous one(s). The strong increase from the rehearsals compared to the base presentation is largely due to the consistency both in transitions and the speech recognizer (mis)matching certain phrases.

Since the purpose of the system is to initiate a slide transition at the appropriate time, the position of the speaker in the presentation is tracked. A window of the current and the next slide is monitored in order to determine the moment the transition should take place. Assuming presenters will address each of the elements present on a slide; this tracking can be quite accurate with the rehearsal system. It also imposes a restriction on the system: a presenter is more or less forced to narrate a presentation in roughly the same way each time, since the system is trained for (a set of) certain phrases.

2.2 Adapting the presentation

Opposed to the semi-fixed narration that a system as Jabberwocky imposes and the fixed linear fashion in which slides are presented by most presentation tools, Moscovich et al. (2004) investigate a different approach: adapting the presentation to the current needs of the presenter. These needs can change throughout a presentation, especially when the interaction with the audience is allowed. Temporal constraints placed upon the presentation can raise the need for changing the speed or order in which the slides are displayed.

Moscovich et al. (2004) provide a presenter the ability to design a set of possible presentations on any given set of slides. Since a presentation is typically constructed for a set of given time and space constraints, a set of presentations on the same topic can be designed, ranging from a five-minute abstract to a full lecture.

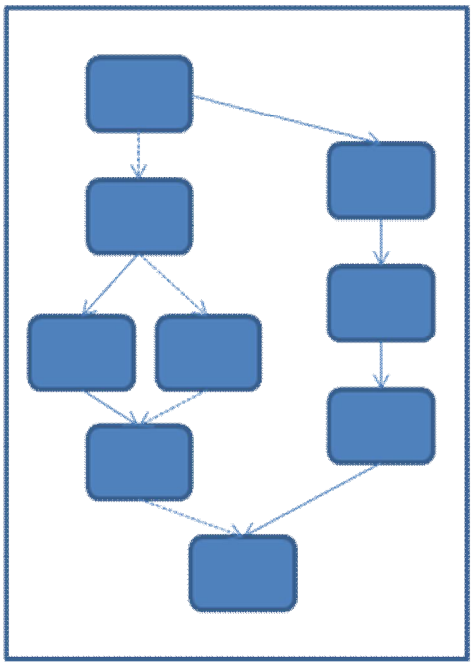

Figure 2 Example of a presentation with multiple paths. The presenter can follow any path depicted by the lines in order to meet the requirements for the current presentation. The dotted lines are an example of a taken path.

Given a fixed set of slides, the presenter is presented the ability to prepare a set of presentations based on those slides. This can result in a set of distinct presentations, each for its own high level purpose. It also allows the presenter to adapt the presentation on-the-fly. Moscovich et al. (2004) constructed and tested a tool that allows navigation through a presentation in more ways than just linear. These customizable presentations allow nested sub-paths, presenting the presenter the opportunity to take a certain branch whilst presenting, thus satisfying changing temporal constraints or perhaps the changing interest or focus from the audience. Figure 1 schematically depicts a set of slides resulting in 3 possible presentations; the chosen path is indicated by the dotted lines.

These customizable presentations are a great step towards adapting a static presentation to the current needs during a talk. It also voids the necessity to create a distinct presentation for every high level intent, since the tool can be used to construct another path through the slides, which basically is a distinct view on the set.

2.3 Generating presentations

So far, the presentations all have been designed in detail by the presenters. The higher intention of the presenter, giving a lecture or a short talk, can be conveyed in a rhetorical structure. The presenter is required to translate this high level intent structure directly into a final presentation, for each intent. Rutledge et al. (2000) have investigated generating presentation structures based on rhetorical structures. The high level intent can be encoded in a set of spatial, temporal and navigational constraints, after which these constraints are sought to be satisfied when constructing the final presentations.

Presenting information with a given set of constraints is something that is already applies in everyday life: HTML defines the layout of data without explicitly prescribing the final presentation. On each interpreter, whether it is a browser at a certain resolution or a mobile phone with UMTS, WAP or any other means of connecting to the internet, the same data is interpreted and translated into a final presentation.

Rutledge et al. (2000) define the constraints in the dimensions space, time and links. They construct a methodology, which allows the expression of high-level spatial and temporal constraints, as well as linking. These relations are often relative to other elements within the presentation space, which allows the final

rendering to take place on a great variety of media. The constraints cannot always be satisfied: overflow of spatial or temporal constraints is something that will very likely occur on different presentation media. It triggers an overflow mechanism, which uses other constraints to try and satisfy the overflowed constraints. For example, if a construction of information elements results in a spatial overflow, temporal constraints can be applied to try and solve this. Instead of displaying the information, taking the spatial constraints in regard, on one screen, multiple screens can be used if the temporal constraints are not violated. This compensation mechanism is an important factor in satisfying the constraints defined in a presentation space and allows a greater set of presentations that satisfy the constraints set as a whole.

Figure 2 illustrates the generation of content. Based on a repository with elements, and a set of spatial (x and y) and time (t) constraints, a set of potential presentations is generated that satisfy these constraints. The user can then select the desired presentation and make the final changes to it, if needed.

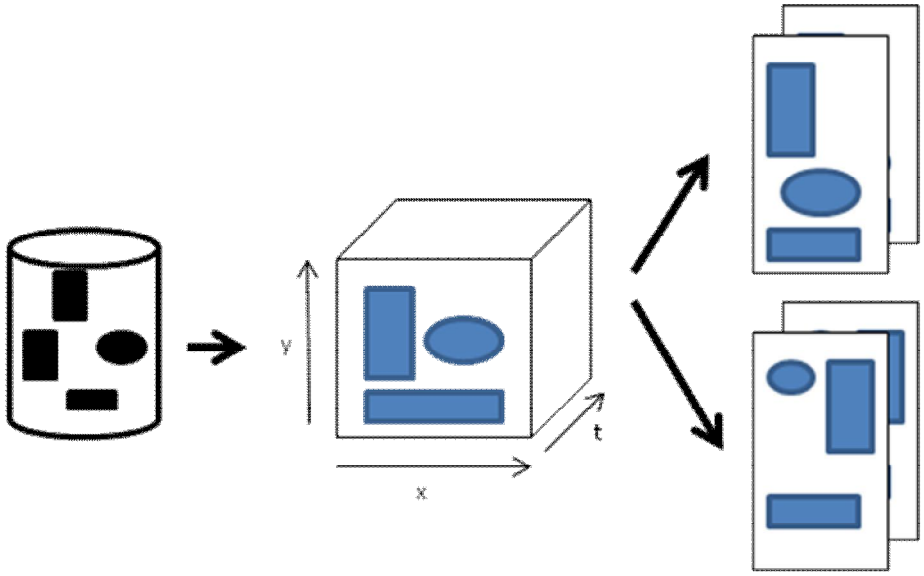

Figure 3 Given a repository with content elements, and set of spatial (x and y) and temporal constraints (t), the system generates a set of possible presentations satisfying these requirements.

The research of Rutledge et al. (2000) is applicable to all presentation media, not just the common presentation used during a talk or lecture. However, applying spatial and temporal constraints on an information repository is the high level intent of any presenter. Talks for different audiences on different presentation media result in different sets of constraints. The result of applying these results is a set of presentations, each suited for a high level intent.

2.4 Standards

One last aspect that deserves our attention is standards. Knowledge is distributed through a variety of media, amongst which the presentation. Over the last decade, digital presentations have been based on the same propriety file format of mostly PowerPoint. An abstract description of a presentation, allowing for (tagged) multimedia content, and an open standard would allow for a greater variety of presentation tools, and tools to develop the presentations.

One of the earliest reference models has been designed by Halasz and Schwartz (1994): The Dexter hypertext reference model. It basically defines three different layers: the run-time layer, the storage layer and the within-component layer. Each layer is designed with fixed sets of operations, allowing the user to design and construct a system in this model. The run-time layer describes the realization of a presentation based on the information stored in the storage layer. The storage layer conveys all the components that can be used in the presenting environment and allows the precise definition of these components. The within-component layer defines the

anchoring: the mechanism for addressing or referring to locations or items within the individual components (Halasz and Schwartz, 1994).

This approach strongly focuses on links between information elements and expresses the data in an XML like language. This model puts emphasis on the hypertext relations between individual components and is thus less applicable to the plain presentation environment. However, the separation of content, relations and run-time designing has had a great influence on a variety of presentation systems.

Bordegoni et al. (1997) proposed a standard reference model for intelligent multimedia presentation systems. They recognize the need for such a reference model from the rise of multi-media content. They view multi-media not as a single entity, but as composition of multiple information elements. The final presentation of the multi-media content should be described in more than just the final product. They acknowledge several projects are initiated to do that purpose, but a standard reference model does not yet exist.

This standard reference model provides a common ground that describes both the information elements themselves, as well as the modalities in which the information is conveyed and the media in which it is presented. Presentations can be expressed by this means as well, generalizing its content to a level in which it would allow interaction with a variety of other media. It also enables the construction of generic information structures that can be adapted to a great variety of present and future media. The generation of presentations described in the previous section is one of the possible applications of this standard.

3 Presentations

This chapter provides an introduction to presentations. The most common structures and navigational issues will be shortly discussed and referenced against the presentations of the MLMI of 2004 (IDIAP research institute, 2004).

3.1 History and context

Exchanging information is one of the fundamental interactions between humans. Distributing pieces of information can be achieved in two ways: actively or passively. The most common examples of passive knowledge distribution are books. Information is written down and thus stored in an information carrier (e.g. a book or a magazine). Whenever the need arises, anyone can consult the information carrier and extract the desired piece(s) of information.

Active knowledge distribution requires interaction between the one(s) holding the information and the people requesting it. The information can be (partially) contained in an information carrier and is managed by the one(s) holding the desired information. The knowledge is to be distributed from the information carrier to the requesters. This process is often realized by giving an (oral) presentation. The person holding the desired information broadcasts the information and the audience receives it. The presenter is more often than not assisted by some passive information carriers such as figures, tables and text, often contained in a presentation.

Transparent overhead slides and overhead projectors as well as chalkboards have supported this form of knowledge distribution for several decades. A presenter carried with him (or her) a set of transparent overhead slides that were placed sequentially on an overhead projector, thus projecting it on a board or screen. Displaying another slide was initiated by removing the current slide, selecting the appropriate next slide from the set and positioning it on the projector. An alternative presenting aid was the slide projector, allowing the presenter to project small slides on a screen. Navigating through the set of slides occurred in the same manner as with the transparent slides: the presenter consciously selects the desired item from a given set and inserts it into the projector. The major downside of displaying information this way is that information once written down is hard to change or remove. Using multiple overhead slides at the same time, or writing on the slides during the presentation are the options a presenter had of adapting the information during the presentation.

Since the nineties, computers equipped with software presentation tools such as PowerPoint typically replaced the overhead projector and slide projector in assisting the presenter. The analogue skudes are replaced with digital slides, which convey the same information elements such as text, enumerations and figures. Additionally, multimedia content such as movies, pictures and demonstrations can be included as well. Modifications and other operations are possible on both analogue and digital slides, but are simplified in a digital environment: insertions, deletions and changes can be made, thus supporting adapting the presentation on the fly. The information has become volatile; changing the information has become as easy and accessible as reading it, and changes can be reverted at any given time.

3.2 Structure

Given the notion that presentations are given with the purpose of transferring information, this goal can be aided by giving presentations a certain structure. Many presentation guidelines recommend presentations to start out with an introduction and a layout of the presentation, followed by the body of the presentation and ending with conclusions and recommendations:

- **Introduction**. A presenter should introduce both himself and the subject. Background information on the presenter could include relevant experience in the domain of the subject of the presentation, as well as global achievements in that field. Even under the assumption that the audience would have some

knowledge of the presentation domain, a short introduction can be used to both introduce the basic concepts and/or terminology as well as setting the boundaries for the topic.

- **Overview**. In order to inform the audience how the presentation will unroll, the introduction is often followed by explicitly stating the exact structure of the presentation, often contained in a table of contents or a similar overview. This allows the audience a clear view on the issues to be addressed and can provide a sense of progression throughout the presentation.

- **Body**. The body of a presentation consists of addressing the topics indicated in the introduction and/or table of contents. The related information of each of the indicated topics is contained on one or more slides. When starting a new topic, presenters often explicitly state that they are, sometimes referring to the table of contents or by displaying a short overview of the subtopic. The terms mentioned in the table of contents frequently reappear as titles of the slides containing that topics content. This repetition can serve two purposes: it allows the audience to keep track of how far the presentation has progressed, and second, explicitly state the topic that is currently addressed.

- **Conclusion**. After the body of the presentation, some sort of conclusion and/or some recommendations are presented. These can often accompanied by a short summary, especially if the presentation tends to be a larger one. After presenting the conclusions and the recommendations, the presenter typically ends the presentation with asking the audience for questions.

We assume presentations to follow these top-level guidelines. Apart from the fact that every guideline or presentation aid suggests this layout, from a logical point of view it is the structure that makes a coherent and continuous whole.

3.2.1 MLMI Recordings

No exact numbers are available, but according to presentation guides and good practices, a well-prepared presentation should follow the following ratios of introduction, body and conclusion: 10–20%, 60–80% and 10–20% respectively. The MultiModal Interaction Machine Learning Algorithms, or MLMI (IDIAP research institute, 2004) workshop publishes the presentations that are given during the workshop. Considering most of the presentations are given by well-experienced presenter, we will use them as a point of reference.

The recordings of the 2004 workshop support the notion of structure: ratios of 11%, 76% and 13% were found for introduction, body and conclusion, respectively.

3.3 Building blocks

As well as the presentation as an entity, the individual slides contain a certain structure as well. This section will provide a quick glance at the building blocks of presentations.

Textual elements

Slides often contain text in some way. Stripped from additional layout issues, it is the most common information carrier available. Text can either be structured or unstructured. The former is often displayed in (un)numbered enumerations or lists, the latter in paragraphs or sentences. Due to the limited availability of space, the presenter has to carefully formulate any text he wishes to put on the slide. This most probably results in having information-dense text.

Tables, figures and other graphics
Although plain text can be used to display any information required, there are better structures to display (structured) data. Tables are used to present relations between two or more entities and provide a clear overview for the audience. Graphical data such as graphs or images are placed in figures.

Multimedia content
Additional multimedia content can be embedded into presentations. Movies, sound files, even (links to) demonstrations of software can be incorporated. The actual handling of multimedia content is often done by third-party software and is usually not embedded within the presentation. Therefore, we consider multimedia content not to be a part of the presentation itself and as such will be discarded for all purposes.

3.4 Navigation

The most intuitive way of navigating any ordered set is sequentially selecting an item, processing it and moving on to the next item. Since a presentation is most likely given in a linear fashion, the presenter will start at slide one and continuously select the next slide until the last slide is reached. Presentation software products let presenters initiate the slide transitions by having them carry out a single action.

This action can be pressing a key on a keyboard, pressing a mouse button or sending a signal to the computer by almost any gadget. However, initiating the action twice (or even more often), triggers the software to repeatedly display the next slide. This may have an unwanted effect, since the wrong slide can be displayed on the background. Carrying out an action more than once can easily be triggered by human aspects, such as nervousness or agitation, as well as technical aspects such as an unstable signal or interference on the frequency. This potentially leads to unwanted behaviour of the system and raises the need of correctional actions, such as selecting the previous slide.

Another aspect of presentation navigation is the addressing of questions from the audience. The presenter may want to directly navigate the presentation in order to display a certain slide. This navigation process requires a detailed knowledge of the slides, since only the presenter knows if and which slide contains the desired information.

These aspects combined raise the necessity of at least the following navigation abilities:

- Selecting the next slide
- Selecting the previous slide
- Selecting a specific slide anywhere within the presentation

Most presentation software products support these actions and thus allow the user to actively navigate the presentation.

3.4.1 Navigation within a slide

In digital presentations, displaying a slide no longer necessarily is a singular action. Slides can contain animations, and the slide can be made visible progressively by adding content based on a timer or an interaction with the presenter. Making content visible on a slide is executed the same way as navigating slides is implemented: with previous and next commands content can be made visible or removed again.

For all intents and purposes, we consider displaying the next slide within this thesis as a singular action, and will address implementing speech for partial slides during designing and evaluating the presentation assistants.

3.4.2 The speech dilemma

Using speech in order to navigate a presentation can be done in two ways: actively and passively. When a presenter utters a command to execute a navigational action (e.g. 'Show the next slide'), speech is actively used to navigate through the slides.

A presentation assistant that listens and, to a degree, understands what the presenter is saying could be developed to display the correct slide; speech is then used in a passive way. It does introduce a fundamental problem: when an assistant decides that the speech of the presenter covers information of the next slide, and decides to initiate the slide transition, it is already too late. Compared to the current situation, where a presenter first initiates a slide transition, and only then starts covering the content of that slide, a delay is inevitable. Minimizing this delay should be an assistant's priority.

Another way of implementing speech-driven navigation is by matching the speech to the current slide, and initiating a slide transition, if and only if the information elements of the current slide have been discussed. The largest downside of this approach is that skipping content is no longer an option, and missed elements can only be compensated for by explicitly listing them in order to satisfy the requirement of completeness.

Both passive approaches are based on the assumption that a presenter's speech is related to the content of the slides.

3.5 Relevance measure

The slides of the presentation are supposed to support and aid the presenter in giving the presentation. The content of the slides should be related to the current speech of the presenter in order to fulfil that purpose. This relation can be very strong and exact if the presenter is reading aloud the contents of that slide or can be a bit weaker if the slides is a mere example of the concepts the presenter is talking about. Whether this relationship is strong or weak, we assume for this thesis that: **There is a relation between the speech of the presenter and the current displayed slide.**

In order to interpret how strong this relationship is, we need to define a relevance measure. This measure relates the utterances of a speaker to the content of the slides by examining their overlap. We introduce S_i, $1 \leq i \leq \#slides$ to be the set of words and other content contained on the i^{th} slide of a given presentation and P_i the speech as a sequence of words uttered during the display of slide S_i. Since both are sets of words, let R_i be their overlap using the following definition:

$$R_i = \frac{\#(S_i \cap P_i)}{\#S_i}$$ Equation 1 Relevance of a slide

This measure can easily be extrapolated to assigning a value to the overlapping speech of a presenter and the content of the slides for a whole presentation by enumerating over all slides within that presentation and averaging the totals:

$$R = \sum_{i=1}^{\#slides} \frac{R_i}{\#slides}$$ Equation 2 Relevance of a presentation

This equation expresses the content that has been used exactly as on the slides during the presentation. Please note that the interpretation of what a speaker is uttering is not taken into account here, so the actual relevance measure can be significantly higher than the outcome of equation 2.

3.5.1 Existing numbers

The MLMI Recordings (IDIAP research institute, 2004) showed this relevance measure exists. Analysis of four (representative) presentations of the recordings show that the majority of the text (or an equivalent textual representation) contained in the slides is uttered literally during the presentation. Assuming we can view these recordings as representative for well-prepared, structured and presented presentations, it strengthens the assumption that the relationship between speech and the slides is valid.

Note that the boundaries in the speech are assigned manually during the analysis of the MLMI recordings. Longer pauses in between sentences typically indicate a topic change. Along with the assumption that slides contain unique information, a topic change is typically accompanied by a slide transition. When this explicit pause is absent, this boundary between slides has been assigned arbitrarily. The assignment of boundaries should be automated in order to prevent the presenter from actively interacting with the presentation.

The presentation assistant will use speech recognition in order to transform the speech of the presenter into its textual equivalent, as well as transforming the content of a slide into its textual representation. First, we will discuss the basics of speech recognition and advance into keyword spotting and grammar-based recognition in the next chapter, after which the exact approach is discussed in chapter 5.

4 Automated Speech Recognition

This chapter briefly glances at the statistical approaches used in Automated Speech Recognition (ASR), after which keyword spotting and grammar-based recognition is discussed.

4.1 From audio to speech

Speech is essentially nothing but a series of sound waves. A speech recognizer transforms this signal into a sequence of words. This process can be broken down to the following phases (Rietveld and Heuven, 2001):

1. The speech signal is captured by a microphone or a similar recording device.
2. The analogue sound waves are transformed into a digital equivalent.
3. The digitized speech signal is divided into sequences of phones.
4. Using a pronounce dictionary, several recognition candidates are generated.
5. The word with the highest probability is chosen. Syntactic analysis and the aid of advanced language models (LM) are common tools for calculating the probabilities.

A lot of information can be obtained from just the waveform, such as presence of voicing, stop closures and fricatives (Jurafsky and Martin, 2000). Other aspects require a different representation: spectral features. These are based on a Fourier transformation of the waveform and represent the different frequency components of the wave. Most modern speech recognizers smooth the spectrum using Linear Predictive Coding (LPC). This algorithm makes it easier to spot the spectral peaks, better known as formants. The LPC spectrum can be represented by a feature vector, containing two features for each of the five formants and two additional features for spectral tilt (Jurafsky and Martin, 2000). A phone is identified by aspects of the formants, enabling recognition of phones and thus syllables and words.

The task of any ASR system is to find the most probable sentence $\hat{W}$ by computing $P(W|O)$ for every possible sentence W given an acoustic observation O:

$$P(W|O) = \frac{P(O|W)}{P(O)}$$ Equation 3 Acoustic observation

$P(O|W)$ is the likelihood that a certain observation is made given the sentence W and $P(W)$ the prior probability that W is obtained using the language model(s) (Jurafsky and Martin, 2000). Since the acoustic observation O always has the same probability and only the relative values are relevant, 4.1 reduces to calculating $P(O|W)P(W)$. Maximizing this probability over all words results in the most probable sentence $\hat{W}$:

$$\hat{W} = \max_{W \in L} P(O|W)P(W)$$ Equation 4 Sentence probability

Since the speech of a presenter needs to be recognized into its textual equivalent, these formulas provide the basics for recognizing that speech.

4.2 Language models

Recognizing speech is more complicated than gathering the phonemes and doing a lookup in the lexicon. Often some words are more likely to occur then others; sometimes the outcome space is limited to a (small) subset of the lexicon (e.g. keyword spotting and grammar-based recognitions; see below). When recognizing natural or conversational speech, statistical models are used to calculate the outcome.

This statistical model is often conveyed in a hidden Markov Model, or HMM. This depicts a directed weighed graph defining the possibilities of transforming phonemes into words. Another common structure is an N-gram. Here, N stands for the number of previous observations to take into account when calculating the current observation. State of the art speech recognition systems have up to N=4 grams language models.

For further reading, we refer to Jurafsky et al (2000)

When developing a presentation assistant, using a relatively small and closed vocabulary language model such as keyword spotting or grammar-based models promise to yield better results, which will be investigated in the next sections.

4.2.1 Lexicons

Since acoustic observations have to be transformed into words, knowledge of how words are constructed is required. A lexicon is a transcription of acoustic observations unambiguously to words. A word can have multiple acoustic observations, but each sequence of observations can lead to only one word.

Lexicons use phonemes to depict the acoustic observations. The phonetic alphabet comes in a lot of varieties, of which Sampa (UCL, 2009) is perhaps one of the more widely accepted ones. It has adaptations for over 25 languages and has been extended to X-SAMPA to be language-independent.

Lexicons define the outcome space of transforming acoustics into words. Language models are used to determine the likelihood of the translation process, but a word not defined in the lexicon that the language model uses results in the impossibility of recognizing the words. Uttered words not in a lexicon are often referred to as Out-Of-Vocabulary (OOV) words and are typically recognized as words that have a similar phonetic transcription.

4.2.2 Keyword spotting

The isolation of words in continuous speech is often referred to as keyword spotting (KWS). In the late 1980's, customers of a telephone company were able to use a speech-driven telephone operating system, which could assist them with a number of tasks, such as requesting a collect call, asking for an operator, etc. (Wilpon et al., 1988). It became clear that customers were reluctant to utter only the required words, but rather placed them in fluent speech. Hence, the need for searching keywords in continuous speech utterances became apparent. Wilpon et al. (1990) show that hidden Markov Models can be used to identify keywords in unconstrained speech, which resulted in a recognition rate of 95,1% of spoken words in fluent speech spoken over long-distance telephone network.

When using keyword spotting, the ASR requires a dictionary containing all the possible pronunciations of the desired keywords. Each entry in the dictionary consists of a word and its possible pronunciations in phoneme representation. The non-keyword speech can be modelled by unconstrained networks of mono phones, which are called filler models (Rose and Paul, 1990). Several approaches exist to construct filler models, but they have in common that they (attempt to) model non-dictionary words in order to lower the false-positive rate.

4.2.3 Grammar based recognition

Instead of recognizing specific words, the recognition of certain sequences or repetitions of words is demanded as well. In order to define the exact sequences, alternations and/or repetitions, something is needed to define the exact structure. Grammars are the best candidate to convey the desired structure. A grammar consists of a quintuple (N,Σ, P, S):

- N: a set of non-terminal symbols
- Σ: a set of terminal symbols, disjoint from N
- P: A set of production rules of the form A -> B, where A ε N and $B \subset (\Sigma \cup N)$
- S: The start symbol

Grammars can be defined using the same operators as regular expressions. Take for example a speech recognition system to substitute a regular remote control for common electronic devices such as televisions, stereo sets and DVD players. It should support the same functionalities as the remote control, such as turning the device on and off and switching to channel between 1 and 99. The first rule of a grammar used for this system should enumerate the possible top-level commands, such as turning the machine on or off and switching to a certain channel. The alternatives can be derived further in additional rules. The example of a grammar for a speech-driven control a set of electronic devices is given in figure 4.1 and is based on the example given by Pellom and Hacıoğlu (2004).

```
S            →  $cmd*

$cmd         →  turn power $power_state [the] $device | $device
                $power_state | [go] [to] channel $channel;

$power_state →  on | off;

$device      →  tv | dvd | vcr | stereo;

$channel     →  $digit | $teens | $tens [$digit]

$digit       →  one | two | three | four | five | six | seven | eight |
                nine

$teens       →  ten | eleven | twelve | thirteen | fourteen | fifteen |
                sixteen | seventeen | eighteen | nineteen;

$tens        →  twenty | thirty | forty | fifty | sixty | seventy | eighty
                | ninety;
```

Figure 4 Grammar for a speech-driven remote control; several devices can be turned on or off with this grammar; or a channel between 1 and 99 can be selected.

The tokens not starting with a dollar sign are the literals in this grammar. Assuming a background lexicon with the transcriptions for all the literals, an ASR is able to recognize the structure this grammar denotes, in this case a set of commands to turn on or off a certain electronic device, or switch to a certain channel.

The example shows a simple application of a grammar-based approach to recognition, the set of potential rules however is virtually limitless in most modern ASRs, allowing for a detailed description and listings of the alternatives.

4.2.4 Finite state grammars

An alternative approach is defining grammars as finite state grammars, or FSGs. In this notation, the grammar is defined using states and arcs:

- S: A set of states
- A: A set of directed arcs of type S->S, with a symbol X.

Depending on the implementation, the following features can be added:

- A start state S_0 and a set of accepting states $<S_{accept}>$.
- A function weight W(S->S) assigning a cost or a probability to each arc.
- A function O(S->S) assigning output symbols to each arc.

In speech recognition systems, the system will load in a start state. The acoustic observations, translated to words using the language model and lexicon are used to determine which arc is a possible transition. Only arcs with symbol X that matches the incoming token are added to the possible set <P>. The next token is parsed and matched against the arcs starting from each state in <P>. When a state in <P> has no matching symbol <X> matching any outgoing arcs, the set is removed from <P>.

This process is repeated until all the incoming tokens and parsed, and the resulting set <P> defines the reachable states given that input.

If the particular system has the ability to define accepting states, it only accepts incoming when there is at least one accepting state in <P> after parsing the input tokens. Typically, this is accompanied by an output symbol, so an external entity can keep track of which path (or rule) has been taken to accept the income.

4.2.5 Filler models

In keyword spotting approaches, the modelling of out-of-vocabulary (OOV) and non-speech sounds such as lip smacking, coughing or throat clearing is done by designing filler models. The first approach is to explicitly restrict the utterances in the template recognition network. Since it can be predicted what will be uttered next under the assumption that no other phenomena occur, high performance is reached at the cost of flexibility. Only explicitly defined sequences are allowed. The next approach allows the ASR to focus on segments of the keywords. Only (sequences) of phonemes are accepted that are defined by the grammar rules, the rest is ignored.

Zhang et al. (2004) describe two ways in which filler models can be constructed. The first is train Hidden Markov Models (HMM) that exactly model extraneous speech prior to using the KWS system.

The other approach is to create filler models on demand, calculating the probability scores for both the filler HMM and the keyword HMM for each text frame (Rose and Paul, 1990). The latter enables the construction and re-use of filler models whenever a grammar is changed.

The latter approach is also the more desirable one for a presentation assistant. The speech will a presenter utters during the presentation will hardly follow a defined pattern to its every phoneme without filler words such as “uh” and “ah”, or any non-speech sounds.

4.3 Probabilistic approach

Given a grammar, it is very well possible that one alternative in a rule should occur more often then another. Jurafsky and Martin (2000) discuss a probabilistic approach to the production rules in a grammar by assigning probabilities to sub-trees in the derivation process of sentences (given a certain grammar). Each production

rule in the grammar is assigned a probability p and thus becomes of the form A -> B[p]. The weight of all the rules given the same start symbol A should sum up to 1.

Calculating accurate values for the individual probabilities is typically an automated task. For example, when using grammar-based recognition to recognize the name of a subject, taking into account the number of occurrences of that particular name provides a good basis.

Another use of probabilistic derivation is to disambiguate sentences (discussed in Jurafsky and Martin (2000)). When analyzing a sentence that can have multiple semantic meanings, the probabilistic values can aid in disambiguating the sentence. An example: the sentence "*Jan ziet Piet met de verrekijker*" (*Jan sees Piet with binoculars*) has two meanings. The first meaning is that Jan views Piet through binoculars; the second is that is Jan sees Piet holding binoculars. Statistically, the former is more likely to have the correct meaning than the second, which can be resolved by assigning a higher probability to the derivation tree that leads to that interpretation.

4.4 Performance

In order to define a generic performance measure for speech recognition, it is essential to first look at the different type of errors that can typically occur during the recognition process. As a first, a certain word can be identified by the ASR, whereas there was no word spoken: insertion. The opposite of this is that a spoken word is ignored: deletion. Last, a spoken word can be wrongly recognized: substitution.

Since multiple errors can occur for a single recognition, the number of insertions, substitutions and deletions between a wrongly recognized word and its correct equivalent in the corpus can be counted. Wagner and Fischer (1974) named the minimum amount of errors the *minimum edit distance* (MED) and provided an algorithm to calculate it. The probability that a certain word matches an entry in the corpus is estimated by minimizing the MED for the entries. Summing these MEDs required for the matching in a certain transcript gives the most common performance measure for speech recognition: the word error rate (WER):

$$WER = \frac{\#Insertions + \#Substitutions + \#Deletions}{\#Words}$$

Equation 5 Word Error Rate definition

An example (taken from Jurafsky et al, 2000) of calculating the WER for a given utterance (REF) that was recognized (HYP):

REF	i	***	**	UM	the	PHONE	IS	i	LEFT	THE	portable	****	PHONE	UPSTAIRS
HYP	i	GOT	IT	TO	the	****	FULLEST	i	LOVE	TO	portable	FROM	OF	STORES
eval		I	I	S		D	S		S	S		I	S	S
Word Error Rate = 100* (3+6+1)/14=71,43%														

Figure 5 Example calculation of a WER. The recognized text (HYP) has 3 insertions, 6 substitutions and 1 deletion on a total of 14 words in the utterance (REF).

Jurafsky and Martin (2000) report that around 2000 the best recognizers for free speech have a word error rate between 20% (Chen et al., 1999)) and 40% (Hain et al., 1999).

4.4.1 Influences

Rietveld and Heuven (2001) investigated a number of factors that influence the performance:

- Bandwidth. If the speech signal is transmitted using a large frequency range, a speech recognizer will achieve a higher performance compared to transmitting using a small bandwidth (e.g. 300-3300 Hz for telephone; 14-44 KHz for microphone).

- Open or closed vocabulary. If the set of uttered words is expected to be a fixed set, closed vocabulary is used.

- Signal parameters. Using formants is just one of the possible ways to parametrize the signal. However, others will not be investigated in this report.

- Corpus size. The amount of available training material is crucial for performance. If the training corpus is qualitative and qualitative very good, performance will improve. Every word uttered that is not included in the lexicon automatically results in an error.

In addition, we identify the following performance aspects as well:

- Clipping. A microphone (or any recording device) is expected to have a maximum input level. If this level is exceeded, the information above that level is lost. The audio signal is flattened at the maximum level, losing all formant information. When computing the phonemes, the required information is very limitedly available, resulting in a difficult recognition process. This symptom is known as clipping and can severely influence the performance of recognition. A great deal of microphones use some form of automatic gain control (AGC) to counter this phenomenon.

- Background noise and non-speech sounds. When the audio signal recorded with a microphone or similar device is fed directly to the speech recognition system, it will attempt to map each sound wave to a known phoneme. Even without diving into that process in detail, it is evident that mapping non-speech sound waves or utterances in the background from other speakers will result in raising false identified phonemes. This aspect can be largely nullified by using pre-processors that analyze and transform the audio signal.

- Weintraub et al. (1996) studied how spontaneous speech differs from other types of speech. They conducted a series of experiments on sentences with identical transcripts that varied in the speaking style and found that speaking style is a very dominant factor in the performance of large vocabulary continuous speech recognizers: performance degraded from a WER of 29% (careful dictation) to 53% (spontaneous speech).

Given a typical presentation environment, we expect the audio to be of microphone quality. The amount of background noise at a typical presentation should be reasonably low, as attendees of a presentation are (supposed to) quietly pay attention, save the incidental interruption.

Given this high quality audio, the speech recognition system is enabled to properly identify the phonemes.

Typically, a presenter prepares a presentation by either rehearsing it a few times, or at least laying down an outline. The presenter will have an inkling on what to say at any given point, but will easily utter filler words such as “uh” and most likely will use different phrasing each time the presentation is given, if repeated at all. Thus the speech of the presenter is in between careful dictation and spontaneous speech. We expect the WER

of a clearly speaking presenter to be between the indicated values at the last bullet, under the assumption that the background lexicon will at least contain transcriptions for every word contained in the presentation.

4.5 Towards developing an assistant

As depicted in this chapter, the availability of high quality audio is an important factor towards developing a presentation assistant. We expect the circumstances of a typical presentation to satisfy that requirement. As the audience will typically silently pay attention and only occasionally interrupt or generate background noise, the recorded audio signal will most likely contain only the soundwaves uttered by the presenter.

With that clean audio signal, speech and language processing technology provide the basic foundation required by a presentation assistant. The next chapter will cover our approach in developing an assistant.

5 Approach

This chapter discusses the approach for creating a presentation assistant. We will start out by defining the limits and context of the presentation assistant and the choice for a particular ASR system. Next, the construction of grammars is described in-depth.

5.1 Overview

The main purpose of the presentation assistant will be to determine the moment a slide transition is in order. It will determine that moment by analyzing the speech of the presenter, recognized by an automated speech recognition system. The ASR uses the constructed grammars for its recognition and the output will be matched against the models created of the slides. In order to design and create the assistant, we will first define the context in which the assistant operates. Next we will move on to the choice of an automated speech recognition system, with the restriction it must be able to support grammar-based recognition. The construction of these grammars is the main focus of this chapter, and will be discussed at length for each operation mode of the system, and the content types typically found on slides.

5.2 Context

We assume the presenter is familiar with presenting, and confident at it. Nervousness often leads to incoherent sentences, unclear utterances, a lot of pauses and an excessive use of breaks like ”ehm”, which may severely interfere with the system’s performance. The presenter has a microphone, which is typically used to boost the speaker’s volume in front of large audiences, but is mandatory for this system. The microphone is connected to a computer running the system and has access to a real-time speech recognition system. Prior to starting the presentation, the presenter allows the system to index the slides, and load the set(s) of grammar(s) into the ASR.

5.2.1 Sub systems

The presentation assistant will be a piece of software that operates on a Windows-based system. It has access to an automated speech recognition that recognizes the speech of the presenter and offers the textual equivalent to the assistant. The ASR is required to support grammar-based recognition.

The assistant has access to the presentation slides, requiring them to be digital. We chose to use PowerPoint as the presentation carrier, since it is the most common and wide-spread software solution for presentations on Windows-based systems, as well as the fact that it allows for other software to directly access its contents via the OLE (Object Linking and Embedding) technology. Network communication allows each of the components to be located on different physical computers, as long as a stable and fast network connection between them is present.

5.2.2 Presenting

Since the assistant aims at removing the manual actions a presenter typically takes to navigate through the presentation, the presenter no longer has access to a device that would allow him to interact directly with the presentation. The process of presenting should not be adjusted in any way other then removing the interactor; presenting should occur the exact same manner as without the assistant. The presenter should be able to use fluent semi-planned speech, as is typical for giving any well-prepared presentation. Since uttering in a clear fashion is already a requirement for a clear interaction with the audience, the speech recognition system’s preference for clear speech should be satisfied automatically.

5.2.3 Intra-slide transitions

With PowerPoint, presenters are enabled to define actions within a slide. Animations, displaying and removing content elements and playing sound files are all accessible to the presenter. These actions can be time based (e.g. triggered after 10 seconds from the start of the slide), or can explicitly be initiated by the presenter.

A speech-enabled presentation assistant would have no issue displaying a slide that contains time based content. Content triggered by actions however introduce a problem. We do not want the presenter to issue a command for every action defined on a slide, as we consider this very intrusive for the presenting process. A presenter would be required to stop the conversational speech, utter a command, and continue to present the slide for every action. Therefore, we expect require the slides to be singular entities, from a display point of view.

5.2.4 Relevance and distinction

The requirement for relevance between the presenter and the content of the presentation has been established in section 3.5. If there is no strong relationship between the speech and the content, determining the slide transition by the assistant will evidently be virtually impossible.

In addition, there is a need for distinction of the slides. If the content of the individual slides show a great similarity, some approaches will perform significantly worse. Relationships between slides such as causality, enumeration sequences or discussing (different) aspects of a certain topic are still allowed, but including slides that share almost the same content is not desirable. We therefore impose that slides have to be unique within any given presentation.

Now that the context of the presentation assistant is defined, we will discuss the choice of an automated speech recognition system.

5.3 Speech Recognition Systems

Automated Speech Recognition (ASR) systems extract the linguistic entities from the speech waves and transform them into a textual representation (Rietveld and Heuven, 2001). This transformation of sound waves into textual representations of speech provides access to text analysis algorithms, which the presentation assistant will require.

There is a great variety of speech recognition systems available, amongst which the following:

- Philips SpeechMagic focuses on document creation from domain-specific speech, as well as template based dictations. The latter offers high performance for domain-specific tasks in for example the medical sector (Philips, 2007)

- Dragon Naturally Speaking allows personal users to interact with their software products (such as Word and Excel, instant messaging and emailing) by voice. The medical and legal business benefit from the adaptations to their fields, although general corporate life has access to the same interaction as personal users (Nuance, 2007).

- IBM's ViaVoice offers client/server solutions for a range of speech-driven applications, such as authentication mechanisms, platform-independent access to information repositories and management of voicemail, email and faxes (IBM, 2007).

- Nuance OpenSpeech Recognizer focuses on the telephony domain. It supports a variety of languages and dialects and allows for grammar-based recognition. (Nuance, 2008)

- SHOUT is not so much as an off-the-shelf speech recognition system, but rather a toolkit to design, create and adapt speech recognition systems (Huijbregts, 2009).

- Spraak is developed by the ESAT department of the university of Gent (ESAT, 20009) as a part of the STEVIN project Spraak (STEVIN, 2009). It is a large vocabulary continuous speech recognizer, fully adaptable and extensible at every step in the process.

Especially SpeechMagic and Naturally Speaking are extended with an extensive programmable interface, allowing easy use of the systems. They also enable the users to deploy a rich variety of applications that use speech technology. However, these are commercial systems, requiring (expensive) licenses. Its core technology is protected, making it impossible to control the exact parameters for speech recognition. These factors render these speech recognitions less attractive for the use in this thesis.

Given that Spraak is relatively new and unexplored within the Human Media Interaction department, we chose Spraak as the speech recognition system for this thesis.

5.3.1 Spraak

Spraak is a large vocabulary continuous speech recognition system. It has been developed by the ESAT department of the University of Leuven (ESAT, 2009) within the STEVIN project SPRAAK. The first release was in 2006, after which development continued to today.

Spraak is developed in such a way, that all libraries are externally accessible and comes with extensive API documentation (Spraak API, 2009). Its architecture and implementation have been developed in such a way, that every step in the recognition process is influenceble or extendable.

Models

The Radboud University developed a generic acoustic model for Dutch in Spraak, which will be incorporated in this thesis. As it is a generic acoustic model, no explicit gender detection is used.

Accompanying the acoustic model is a 4-gram language model based on parts of the Corpus Gesproken Nederlands (corpus conversational Dutch) (CGN, 2009) and the Twente News Corpus (TwNC, 2009). The words used in the language model are available in a lexicon, which will be used as the basis for generating phonetic descriptions of the content of the slides.

When words are not in the available lexicon, the experimenter provides a phonetic transcription before constructing the models.

5.4 Command types

The remainder of this chapter discusses several approaches for creating a presentation assistant. All these approaches have in common, that they require the existence of a certain command structure: the presentation system has to perform actions given the speech. Different types of actions can be distinguished, which we will define here. The complete set of commands will be defined as C and will contain the following types:

- **Next()**. The next slide of the presentation is to be displayed. If the last slide is reached, this action will not be executed.

- **Previous()**. The previous slide of the presentation is to be displayed. If the first slide is selected, this action will not be executed.

- **Jump(*x*)**. Slide number x will be displayed. This requires $1 \leq x \leq$ #slides.

Any developed system should be able to support this commands to allow for direct interaction with the presentation.

5.4.1 Grammars in Spraak

In order to support the recognition of commands in speech, we need to define the grammars that correctly recognize the tokens contained in the commands. As described in the previous chapter, several approaches exist to describe grammars.

The support for context-free grammars, or grammars in Breibach Normal Form is limited, whereas finite state grammars are well-documented and ready to be used. For this reason, we choose to describe the grammars in a finite state model.

This imposes some restrictions on the models:

- All the words generated from the content of the slides must be available in the background lexicon with a correct phonetic description. These can be retrieved either from the background lexicon, or will be provided by the experimenter when required.
- We define a single starting state S_0="0" and a single ending state in order to measure if a stream of tokens is accepted by the grammar.

In the remaining of this chapter we will show how the grammars used by the presentation assistant are constructed and extended.

5.4.2 Commands

In order to navigate through the presentation, the presenter must have the ability to at least access the following commands:

Command (Dutch)	Meaning	Command type
(Toon de) volgende slide	(Show) next slide	Next
(Toon de) vorige slide	(Show) previous slide	Previous
Toon slide <*x*>	Show slide <*x*>	Jump(*x*)

Table 1 Available commands for the presenter

This command structure can be translated into a FSG by starting with one accepting string and gradually expanding the grammar with new rules. The command "toon de volgende slide" is accepted by the following grammar:

Figure 6 Basic next() command grammar

Each arc is accompanied by the symbol allowing the transition from the begin to the end state. Since Spraak has a strong silence detection, we allow for silences between the words making up this rule:

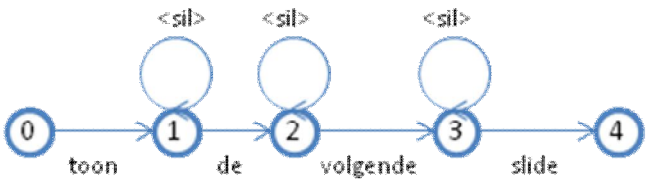

Figure 7 Next() command grammar, now with silences

Spraak uses <s> and </s> to explicitly mark the start and end of a speech segment. These must be included in the grammar as well, introducing a new start and end state. In addition, we want the *Next()* command to allow for uttering "volgende slide" as well instead of always being preceded by "toon de". Spraak allows for arcs without an explicit symbol, called ε arcs. Including these features results in the following grammar:

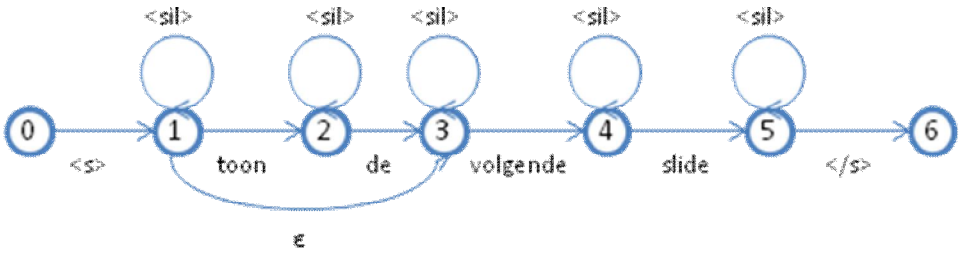

Figure 8 Next() command, as implemented in Spraak's FSG

Adding the other commands is trivial, with the exception of the *jump(x)* command. This command will be implemented by having "toon [slide/sheet]" in 2 states, followed by as many arcs and states as there are slides in the presentation. This way, a presenter cannot navigate to a slide outside the boundaries of the presentation, nor can a false positive recognition to such a slide occur.

5.4.3 Costs

Since Spraak allows for the arcs to be weighed with a cost measure, we assign the following values:

- If an arc is in an accepting line (e.g. a word part of a command), the cost for travelling that arc is set to 0.
- If a silence period occurs in a given state, it will be awarded the low cost of 3. This way, silences are not discarded, but are also not encouraged.
- We do not want to allow more than one silence period in a recognition. This can be realized by setting the fail cost to -5. Two silences would cost -6, which means the derivation fails.
- If the interpreter is no longer able to continue accepting tokens (e.g no token is provided that has an arc defined from the current set of possibles <P>), we do want to allow for restarting the rule, but not influencing the total cost. This is implementing by using the Spraak "fallback" feature, which allows falling back from a given state to another state for a certain cost. We set this cost to be the same as for failure: -5.

This configuration allows for single silences within a single command recognition as well as restarting the grammar if the provided token does not match the arcs leaving the current state. The latter allows us to implement some form of keyword spotting, as the grammar can start accepting at any given time in the speech segment.

5.4.4 Spraak datafiles

A Spraak FSG file has to start with the magic cookie [FSG], followed by stating its name and number of states and arcs. Please note that these numbers must be accurate, or the grammar will not be accepted at all by Spraak, or can display unwanted behaviour (e.g. when the number of defined arcs is lower than the amount of actual arcs).

```
[FSG]
name <name>
Nstate #states
Narc #arcs
Fail_cost <failcost>
accept <acceptList>
fail_cost -5
output nextSlide prevSlide jumpSlide
fb_state 1 -5 2 3 4 5 6 7 9 10 11 12 13 14 15 16
arc 0 1 <s> 0 []
arc 1 2 toon 0 []
(..)
```

Figure 9 FSG file format, as used by SPRAAK

The cost for a recognition to fail is set by fail_cost, and the output defines the list of allowed output symbols. We use the fb_state token to specify that from each state, a fallback to the start state after <s> is allowed at the fail cost.

The file is completed by defining the arcs, each on a new line in the following format:

```
arc <startstate> <endstate> <symbol> <cost> <outputSymbol>
```

Keep in mind that each of the symbols has to be defined in the accompanying dictionary.

In the remainder of this chapter, grammars are constructed as Finite State Grammars, and are implemented using the above file format for Spraak. Where necessary, we will elaborate on the construction of the grammars.

5.5 The first assistant

The basic set functions the presentation assistant has to supply is allowing the user to freely navigate the presentation. With the commands listed in table 1, this requirement is met. The first presentation assistant is equipped with a grammar based on the commands and the number of available slides per presentation.

Allowing the user to issue commands is within this thesis considered to be a fallback mechanism, when the presenter finds himself on a slide that was not expected. It could have been the result of an unexpected transition, and allows the presenter to actively navigate back to the desired slide by issuing one or more commands.

5.5.1 Activation

In TV shows like Star Trek, the characters of the show always explicitly address the computer if they want to issue some command by stating “<sil>Computer<sil>” and waiting for the small bleep indication the computer is ready to receive a command. The activation protocol makes sense, as it indicates when speech is just speech and when speech should be interpreted as a command.

In developing the assistant, we will not use explicit activation. As an alternative for direct activation, we have included the words ‘sheet’ and ‘slide’ in the command structure in order to distinguish the commands from normal conversational speech. These are words unlikely to be uttered in situations other then navigating, and if presenters use the system knowingly, uttering a command unintentionally is unlikely to occur.

The presentation assistants we investigate in this thesis have to incorporate the content of the slides. In the next sections, we will explore transitioning on different types of content.

Please note: each of the following approaches requires parsing (parts of) the contents of each slide. This process is elaborated after introducing the approaches, in section 5.9.

5.6 Transition on titles

In order to advance beyond the necessity to issue commands, we use the notion that speakers often use the titles of their slides to indicate the topic they are discussing. When actively initiating a slide transitions, presenters find themselves uttering the title of the next slide to gather their thoughts before they continue the presentation.

When we apply this notion to our presentation assistant, we need to assert that the titles of the slides are unique. This is typically the case in presentations, as slide titles are commonly used as topic indicators. When using the presentation assistant, we assume the presenter is aware of this notion and the presentation will have distinctive and unique slide titles.

There are two distinctive methods to implement this type of operating. We can either construct a single grammar containing all the slide titles, or construct a set of grammars, each containing a grammar for initiating the next slide. The major difference between these two approaches is that the former allows jumping from slide 1 to slide 3 if the title of slide 3 is uttered, instead of that of slide 2. Both approaches will be elaborated in the next sections.

5.6.1 Access to all slide titles

The first alternative is to construct a single grammar, containing a rule for each slide in the presentation. Each of these rules would consist of the slide number, combined with the exact sequence of words of the title. The start rule *S* will be expanded with $n = \#Slides$ non-terminals $Slide_i$, $1 \leq i \leq n$. Each of the individual rules will have the following form:

$$Slide_i \text{ -> Title of } Slide_i$$

In a finite state grammar, this results in adding a sub model from the starting state to the endstate. Within a title, we do not allow for silences, thus resulting in sub models of the following form:

Figure 10 Creating a grammar rule for titles

It is clear that at any given time the assistant will jump to a slide x if the title of that slide is recognized. This may prohibit the presenter from giving the presentation in a linear fashion. It does however allow compensating for errors. If the presenter unintentionally utters the title of a slide, the assistant will display that slide. A correction can be made by either issuing a direct command, or including the title of the desired slide in a sentence.

5.6.2 Enforcing linear presenting

When the presentation has to be given in a strictly linear fashion, the grammar should enforce that notion. When the first slide is displayed, only the title of the second slide should be encapsulated in the grammar for recognition purposes, not the other titles. However, after the title for the second slide has been recognized, the need arises to include the title of the third page for recognition, and remove the title of the second slide. A single grammar used for grammar-based recognition cannot ensure this principle. A possible solution to this problem is to dynamically load grammars into the ASR system during the presentation. That way, the current grammar only contains the rules required to display the next slide, not that of any other slide.

Spraak allows dynamically loading grammars for grammar-based recognition. Using the API functions, changing the grammar can be realized at any given moment when Spraak is run. Each grammar will consist exclusively of the commands model, along with a rule to display the next slide. If slide 1 is displayed, grammar S_2 is loaded into the ASR. When transitioned to slide 2, grammar S_3 is loaded, etcetera.

This approach imposes a restriction on the presenter: the presenter must read the slide title aloud clearly in order to display the next slide. If the system misses a transition, the presenter cannot continue presenting without interfering by issuing an explicit command. So, although the performance in terms of error rates is probably better, the severity of an error increases.

5.6.3 Topic clustering

Any presentation can contain multiple topics. Sometimes, these topics are made explicit by sharing the same title, optionally identified by a sequence number. This introduces the situation that when a presenter is discussing a slide within such a topic, the speech will most likely contain the words that match the title of the next slide and thus trigger a slide transition.

This behaviour is inevitable when the slides share the same title (or content for that matter; see the next section). Within this thesis, we ask of the presenter to design the presentation in such a way, that the slides are distinguishable for the presentation assistant.

5.7 Transition on content

Initiating transitions on the titles of slides can be seen as a form of issuing a form of commands. The presenter is to some extent required to utter the exact words of the title in order to initiate the desired transition. When the presenter is unsure of the title, or mispronounces it, the assistant will refuse to initiate the transition. A direct command can still solve the risen problem, but reinstates the necessity to actively navigate the presentation.

Instead of just using the titles of the slides, we can expand the trigger for transitioning to all the content found on the slide. This raises issues with parsing the slides, but this will be addressed later on, in section 5.9. For now we assume that a textual representation of the content found on slides exist. This textual representation can be used to construct the grammars needed for the ASR.

5.7.1 Grammars

We will construct a set of grammars again, as opposed to including all the rules in one single grammar. This could easily introduce overlap between slides, as some elements might occur on more than one slide, e.g. the title of slides could reappear in the table of contents. Determining which slide to display would require additional rules for prioritizing and disambiguating. However, if we assume the linear presenting paradigm is still effective, a grammar would only have to contain rules for displaying the next slide, and still support the issuing of commands. We can thus continue on the approach described in the previous section, and construct a set of grammars for a given presentation.

5.7.2 Parsing content

Each grammar still has to support the commands, thus has to include the command model. Similar to the previous approach, submodels will be constructed to contain the content of the next slide. The following heuristics are applied generating the models.

Footnotes

PowerPoint offers presenters the option to assign notes to slides. These are used in particular printings of a presentation, or can be used by the presenter as reference cards during the presentation. In order to translate the latter paradigm to speech recognition, we will include the notes section of each slide when generating the

grammar for that slide, having the notes parsed in the same manner as the rest of the content. The resulting submodel is added to the grammar for that slide, allowing the presenter to have the notes section of a slide contain keyphrases, on which a transition can be iniated.

Enumerations
Enumerations consist of short sequences of words, each preceded by some sort of bullet. The text from these sequences can be used to construct the rules. The sequences often consist more of keywords then grammatically correct sentences. Since presenters probably will utter the keywords, this does not pose a problem. The text of each bullet is parsed and presented as an alternative in the grammar.

Tables and figures
Presentations often contain non-text elements, such as tables, figures, pictures and formulas. Parsing pictures and figures into a textual equivalent is impossible without the aid of advanced domain knowledge and/or character recognition software. Formula's often lean heavily on the domain they are a part of and tables possibly represent data of a certain domain. In order to enable the assistant to link a grammar rule them, we encourage the presenter to add (sequences of) words to these elements, enabling the system to create grammar rules. These keywords can be placed in the *Notes* section of a slide (see above). It is not necessary to specifically relate them to a specific information element, since the tracking mechanism is restricted to determining the relevant slide, not to elements on a slide.

The resulting models are added to the command models and are considered a single grammar.

5.7.3 Prioritizing content

When looking at a slide, humans can distinguish between elements that can be considered a key element to that particular slide and others that might just be included for aesthetics. This would mean that a slide transition based on one element can be considered a correct one, whereas based on another (e.g. a footnote, a single common word) can be depicted as a false transition. When analyzing the slides in order to construct the grammars, this notion is lost. Within the scope of this thesis there is no room for prioritizing content. It would require tremendous domain knowledge, or generalized heuristics that could just as easily be falsified in other situations.

We therefore chose not to attempt prioritizing generating the content rules based on domain knowledge, or any other information inherent to a specific presentation.

5.8 Parsing text

Slides contain often more than just plain text; pictures, formula's and tables can be included. In order to construct the grammar rules for this content, parsing them into a textual equivalent is required. This section describes the different types of non-textual content and how to translate them.

5.8.1 Numbers

Numbers on slides are often in their direct form, e.g. 10 or 1430. In order to correctly recognize a number, they are transformed into their textual form. Some heuristics are used when constructing these numbers:

1. The first series of a certain magnitude are often uttered differently from their successors. For example, 124 is uttered *hundred twenty four* instead of *one hundred twenty four*. This holds true also for thousand, million and billion.

2. Numbers between 1000 and 2000 can be years. These are pronounced differently from 'normal' numbers: first, the amounts of hundreds is uttered, followed by the remaining part as a normal number. For instance: the year 1987 is pronounced nineteen-hundred-eighty-seven. So, an alternative pronunciation

should be generated for these numbers. This holds for all numbers ranging from 1000 to 10.000, so alternative pronunciations are generated for each of them.

3. Large compounded words often do not occur in a lexicon, whereas the small parts are. Since the meaning of a number is irrelevant for recognition purposes, the words are decompounded. For example, the number 1987 will be transformed into nine teen hundred eighty seven.

5.8.2 Abbreviations

Since the available space on a slide is relatively small, presenters tend to use common abbreviations. These should be expanded into their full form as well, since they can be uttered fully. The abbreviations are contained in a list *Abb* of (abbreviation -> transcription).

5.8.3 Symbol translation

A number of symbols is commonly used in presentation, for example the currency symbol €. A list of well-known and often applied symbols Sym is included in the system, containing a list of (symbol -> representation) rules.

5.9 Operating the assistant

When the assistant is started, the first grammar will be loaded into Spraak along with the corresponding dictionary. The ASR is configured to start listening as soon as the loading process has finished. When the presenter starts his/her presentation, the speech signal is channelled to Spraak. Whenever a recognition is made, the accepting state is reached and an output symbol is generated. These symbols are a direct match against the possible command types (prev, next, jump(x)) and can be executed by the presentation assistant.

The appropriate slide is displayed, after which the corresponding grammar is loaded into Spraak again, with the accompanying dictionary and the process starts over again.

5.9.1 Stopping the assistant

This is continues until the presenter explicitly closes the assistant. We chose not to enable this by uttering a specific command. Although the implementation would be trivial, the results of a false positive would be destructive; when the system shut downs the presenter has no means of recovering. We consider this to be highly undesirable and thus exclude such a command from the environment.

Pausing and resuming the presentation assistant could be a possibility, but has not been explored within this thesis.

5.10 Implementation

The presentation assistant has been implemented in Borland Delphi 7 for Windows. The choice for this platform is based on the available knowledge and required components to interact with other systems such as PowerPoint. Direct access to the presentations is provided by an Object Linking and Embedding (OLE) interface of the Office software. It allows direct manipulation of presentations created in Microsoft PowerPoint and basically provides automated actions any user would have access to using PowerPoint. The Delphi implementation of this Visual Components Library (VCL) component for Office 2000 and later has been included since Delphi 6 and is freely usable. Since even the newest versions of Office (up until and including 2007) are backwards compatible, the assistant can be used with most presentations available.

6 Evaluation

This chapter covers the evaluation method used to test the performance of the presentation assistant. First, it briefly discusses the choice for the used method, after which the experiments and their setting are described. Next, a set of performance measures is established and calculated.

6.1 Evaluating a presentation assistant

As we have discussed in the previous chapter, the presentation assistant can operate in different modi, ranging from a basic command driven operation to an analysis of the slides. We want to evaluate how well these systems perform. In order to compare the systems, we need to define a set of performance measures and measure them by conducting experiments.

6.2 Slide transition timing

The main goal of the presentation assistant is to initiate a slide transition when this is deemed appropriate by the system. This moment is defined by a change of topic from the current slide to the next, although not necessarily its successor in the set of slides. Since the presentation assistant uses a model based on the content of the slides, a slight delay is inevitable in most operation modi. If the presenter utters a direct command, this delay is avoided, but in any other case the utterance that triggers the transition is considered part of next slide's topic.

A transition can only be initiated in the context-based approaches when the presenter has uttered one more utterances relating to the next slide. In case of an utterance with little or no topic information (e.g. "*Are there any questions at this point?*"), the transition will not take place. If however, an utterance is directly related to the topic of the next slide (e.g. "*Next, I will be discussing..*" , the transition will take place almost immediately after the intended topic change of the presenter.

This delay is considered acceptable and will not weigh in the evaluation, as it is part of how the system operates. It can be considered a soft performance measure however, as presenters unfamiliar or new to the assistant might become confused and attempt to wait for a transition.

6.2.1 Learning process

Jabberwocky (Franklin et al., 2000), the closest related project, required users to train the system three times before using it in a 'live' environment. The rehearsal phase is needed to abstract keywords and -phrases, which then are linked to the information elements they describe. Each rehearsal, the sets of utterances are enforced and expanded. The biggest advantage of such an approach is that the performance increases significantly when using a trained Jabberwocky. This however also demonstrates its greatest weakness: speaker-specific performance. A trained Jabberwocky operated by another speaker likely results in a drastic drop in performance due to differences in specific utterances and keywords as well as the differences in pronunciation.

The presentation assistant designed in his thesis is meant to be operable the moment it's started. We will not use speaker adaptation or introduce any rehearsal phase.

6.3 Performance measures

Measuring the performance of any given system is identifying key parts of the system (often related to its purpose) and assigning values or scores to how well the system carries out its designated (sub)task. If several systems are to be compared, it is required to measure their individual performances under identical

circumstances. Identifying measurements allows the comparison of the system in an objective way. In order to evaluate the presentation system, we identify the performance measures discussed in the following sections.

6.3.1 Word error rate

The ASR plays an important role in all systems, independent of the exact configuration. Since it is the basis on which transitions can be initiated, it relates directly to the performance of any system. The performance of any ASR can be measured by the word error rate (WER). Its strong point is that it provides a single measure, in which all related models are evaluated, which calculation can be automated.

When determining if the recognition of a sound wave into text is correct, the individual performance of underlying models is considered irrelevant, as only the output text can be used to direct the presentation assistant. Whether the error that occurred originates in the acoustic model, or the lexicon, is of no relevance when determining the slide transitions.

However, its strongest point is actually the weakest point as well. The fact that the entire process of recognizing sound waves into text is condensed in a single measure and should be calculated automatically raises the issue of actual accuracy and interpreted accuracy.

The biggest downside of WER is that it only calculates exactly that: the number of errors in the recognized text compared to a transcription. It measures the performance of the speech recognition system itself, not the larger system the SRS is a part of. More often than not, it's the meaning of the recognized text that matters. Spelling errors and/or errors in filler words are of much less influence on the system at large, whereas the calculated measure may indicate poor performance. Thus we introduce an additional and perhaps more accurate performance measure.

6.3.2 Transition accuracy

When we recall the original research goal, measuring the transition accuracy touches the core of the system. Since the approach is to void the direct user interaction, false positives can raise a confusing situation for the presenter. Measuring the accuracy is basically counting:

- Intended transitions. These are the transitions that are executed when the user intended to.
- Wrong transitions. A transition was intended, but another transition was executed.
- False positives. The presentation assisting initiated a slide transition when the user did not intend one.
- Missed transitions. A transition should have taken place, but the assistant did not initiate one.

As the presentation is unable to incorrectly not initiate a slide transition, since that is the default behaviour, there will be no false negatives relevant for this measure.

Please note that triggering displaying the current slide (e.g. issuing jump(5) when it's the slide that's currently being viewed), will not count as a false positive, as the behaviour of the system remains the same.

6.3.3 Transcriptions

In order to correctly calculate this measure, we will provide transcriptions of every presentation during the experiments. This way, we are able to measure the accuracy against the actual spoken text.

Transcriptions are made with Transcriber 1.5.1 (Manta et al., 2007) and are adjusted for spelling preferences where required. The audio is segmented in turns, using silences as separators. The silence turns are considered turns as well, as false positives can still occur in these periods.

6.4 Experiment setup

Now that a set of performance measures is established, these measures have to be investigated. We will conduct a series of experiments in which test subjects will use the developed presentation assistant. This section describes the setting in which the experiments take place, as well as the constraints imposed the presentations.

6.4.1 Testing the different approaches

In the previous chapter, we identified three major approaches: command-driven, title-based transition and content-based transition. Each test subject will conduct the experiment for each of these systems, leading to a total of three talks per subject.

Participating in the experiments will be anonymous, although the audio and transcriptions of the file are considered part of this thesis. As such, they are provided along with the thesis.

6.4.2 Topic and duration

Simulating complete lectures or in-depths discussions with an audience on a particular topic is undesirable. Different degrees of domain knowledge, as well as the attention span of the audience might interfere with the desire to keep all factors as constant as possible. We expect no great differences in use of the assistant in a small talk then in an extensive lecture, so the test subjects are asked to give a brief presentation, lasting around 5-10 minutes.

Its topic should enable the test subjects to alter the presentation to their own liking, yet does not demand any domain-specific knowledge or introduce huge differences between the subjects. A common situation therefore seems a suitable subject. The process of getting up in the morning and preparing for work has been chosen as a subject for the presentations. Each subject should be able to prepare and (repeatedly) conduct a small 5-10 minute talk about their morning, allowing enough subject-specific changes yet constricting the domain of all the talks.

6.4.3 Preparation

Prior to the actual talk, participants are offered a framework presentation regarding their own specific morning. The duration of the presentation should typically be around 5–10 minutes. Since speakers use an average of two minutes to discuss a slide, the framework presentation consists of five slides. The most common presentation elements are included: enumerations, figures and additional notes.

Subjects are explained the purpose and setting of the system prior to each experiment. When applicable, they are asked to annotate the non-textual elements by placing tags in the Notes field of a slide. We chose to let the participants define their own keywords, so it is more likely their speech will be positively matched against them. The subjects are then asked to adjust the slides to their own situation and are encouraged to introduce new slides. When they have concluded adjusting the presentation to their own situation and liking, the subject can start the system and give the short talk.

6.4.4 Recording speech

During the talks, the subject is asked to wear a microphone, since the assistant requires the speech parsed by the speech recognition system. Luckily, giving a presentation most often requires carrying a microphone to amplify the speech for the audience, making the need for a microphone next to non-intrusive on the presenting process. A Sennheiser PC130 is used to capture the speech throughout the conduction of the experiments. It allows for CD quality of recording, thus meeting the requirements for the recording device.

The experiments are conducted in an isolated environment. Although this allows for the possible audio quality and largely resembles a live situation, it is not performed for a live audience. Typically, the experiment presentations are held for a crowd no larger of the experiment conductor and optionally a couple of listeners who were passing by. Thus it does not resemble the occasional background rumours (if any), nor does it accurately influence speakers who are anxious to perform in front of a larger crowd.

Nonetheless, the similarities to an actual live situation are great enough that we feel confident the data gathered from the experiments is reliable enough to indicate the performance of the presentation in a live situation. The observations and calculate performance measures are a valid indication for a real-world application.

6.4.5 Wizard of Oz

During the first conductions of the experiments, a major problem presented itself: real-time continuous speech recognition was hard to guarantee with the client/server approach. Failing to deliver recognized speech in time for the assistant resulted in unexpected behaviour for the test subject or even complete failure. In order to overcome this problem, we chose to switch to a wizard-of-Oz approach. The experiment conductor acts as the system, simulating both the slide transition system and the ASR. Initiating the slide transitions during the experiments was facilitated by covering up a secondary mouse outside the subjects view and was operated in a way that was transparent to the subject. The subject still wears the microphone, so the speech can be accurately captured for off-line analysis.

This change of approach did not change the setting or preparation of the experiments.

6.5 Results

A total of nine test subjects have cooperated. One experiment is not included in the results, as the speech recorded during that experiment proved to have so many clipped samples and artefacts that recognizing the speech became impossible. In total, 24 presentations were recorded, resulting in almost two hours of speech. Each of the recorded presentations has been transcribed using Transcriber 1.5.1 (Manta et al., 2007) and the text contained in the resulting XML annotation files was extracted for comparison with the recognition result in order to calculate the word error rate.

As stated in section 6.3.3, the resulting aligned speech allows for and will be used for further analysis.

6.5.1 Speaker profiles

As described in section 6.4.1, participating in the experiments will be anonymous. In order to acquire some characteristics of the subjects, they were asked some questions related to their familiarity with presentations and confidence at presenting. In the remainder of this chapter, speakers will be identified by an identifier, assigned in table 2 when required.

Speaker	Gender	Age	Profile
Speaker 1	Male	55	Experienced presenter in both small and larger audiences
Speaker 2	Male	22	Very inexperienced presenter; performs not too well in front of audiences
Speaker 3	Male	24	Reasonably inexperienced presenter
Speaker 4	Male	23	Reasonably experienced presenter; feels comfortable standing in front of audiences

Speaker 5	Male	23	Reasonably inexperienced presenter; feels uncomfortable standing in front of audiences
Speaker 6	Male	23	Inexperienced presenter; comfortable with audiences
Speaker 7	Female	24	Experienced presenter; comfortable with audiences
Speaker 8	Male	24	Experienced presenter; uncomfortable with audiences

Table 2 Participant profiles

Splitting the participants into groups, based on their experience and confidence might have been of use if the number of subjects were greater. It would allow for a more detailed analysis on the influence of the presentation familiarity with regard to the performance of the system, but within the limits of this thesis, this has not been found feasible.

6.5.2 Clipping

During the conduction of the experiments, the used recorder did not use gain control in order to avoid clipping. In addition, the operating system used to facilitate the recording, amplified the signal from the microphone, resulting in severe clipping in some of the recordings, immediately resulting in a large loss of information in the sound waves. As a result, the phoneme detection was severely hampered, resulting in a very low overall quality of the recognized speech.

The first experiments were analyzed with a different ASR then Spraak, SONIC (Pellom et al, 2004), which was unable to properly perform speech recognition on the clipped audio. Spraak on the other hand was able to still perform decently, if the audio contained clipped samples.

As a solution, we will still continue to use the data gained from their presentations, but will point out problems due to the quality of the audio when necessary.

6.6 Obtaining, calculating and interpreting results

Given the presenter's speech, its trafanscription and the grammars constructed with the prototypes, the following process has been followed in order to gain the results described in the upcoming sections:

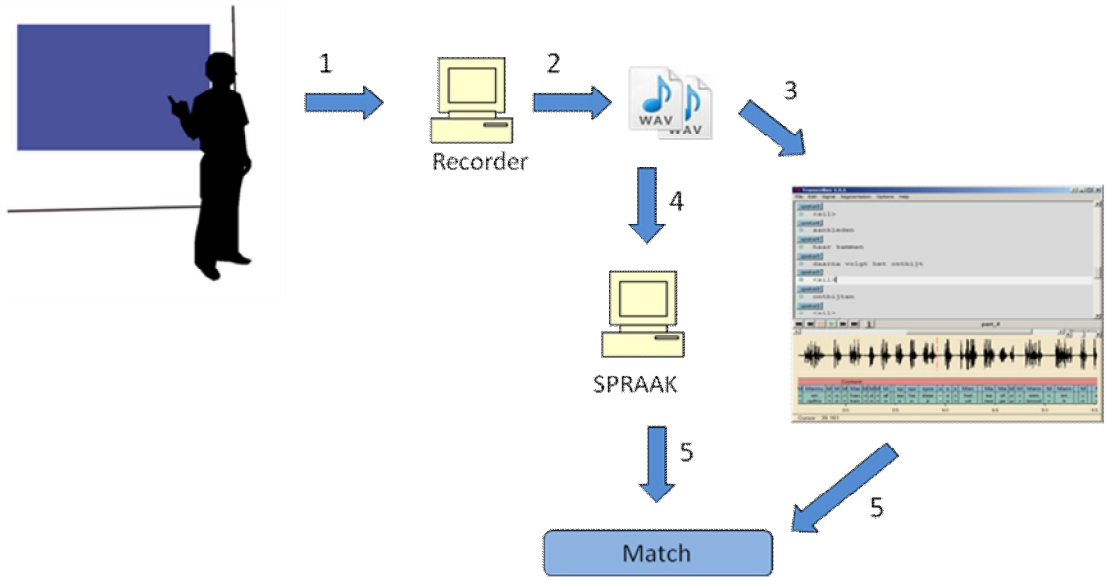

Figure 11 Schematic processing of the results

1. The speech of the experiment is recorded.

2. These recordings are split into turns.

3. The speech of each presentation is transcribed, and adjusted for spelling preferences where necessary. Each turn has its own transcription.

4. For each experiment, a Python script was created in order to perform the actual recognition on the segmented audio. This script writes the resulting text along with its corresponding confidence score in a file.

5. The results are matched against the intention of the speaker by explicitly marking the commands in the original Transcriber file and comparing them to the results. This process is done half-automatically, but manually checked.

This process has been performed for each experiment in order to ensure a single and uniform way to extract the results from the gathered data during the conduction of the experiments. The scripts, data and process are available on request.

6.7 Issuing commands

The first experiment everyone was asked to conduct, was to issue the commands. All commands were uttered twice, during which the subjects were encouraged to use the provided alternatives as well.

The commands were without any exception as such: preceded and followed by at least half a second of silence, and very clearly uttered. The articulation during these utterances was particularly high for most subjects. It was clear to the subjects that they were talking to a computer, thus adjusting their normal speech rhythm and intonation.

Speaker	#utt.	Correct	Wrong	FalsePos	Missed
#1	40	87,50%	12.50%	%	0%
#2	32	93,75%	6,25%	0%	0%
#3	12	66,67%	0%	0%	33,33%
#4	21	90,48%	4,76%	0%	4,76%
#5	17	94,12%	0%	0%	5,88%
#6	25	96,00%	0%	0%	4,00%
#7	23	95,65%	4,35%	0%	0%
#8	20	100%	0%	0%	0%
Average	**24**	**89,17%**	**2,56%**	**0%**	**6,85%**

Table 3 Uttering the list of comands

When analyzing the audio, we find that the majority of the subjects were uttering the commands very clearly. However, given the nature of Finite State Grammar based LMs with a small (and closed) vocabulary in Spraak, just comparing the recognized text with the transcribed text proved to be insufficient as a number of false positives were introduced.

Spraak offers a confidence measure for its recognition results. In almost all of the cases that a false positive was introduced, the confidence measure had a value far below 0, whereas positive recognized commands typically had a score of over 100. Table 3 shows the results for uttering the commands after applying the threshold.

From this point forth, only recognitions that have a confidence of at least 0 are taken into account, all recognitions failing to meet this requirement are discarded.

If nothing else, table 3 shows that closed grammar recognition in a silent environment performs extra-ordinary well.

6.8 Command-based transitions

Using nothing but the command structure rehearsed in the first phase, the subjects gave the second presentation. After making the requested modifications to the presentations in order to reflect their own morning ritual, each subject presented without any additional rehearsal. One subject requested to start over within the first 30 seconds, and did as such. The audio prior to the restart has been discarded.

Speaker	#utt.	Correct	Wrong	FalsePos	Missed
#1	106	99,06%	0%	0,94%	0%
#2	88	93,18%	4,54%	2,27%	0%
#3	34	70,59%	0%	11,76%	17,65%
#4	48	85,42%	4,17%	6,25%	4,17%
#5	104	92,31%	0%	7,69%	0%
#6	-	-	-	-	-
#7	61	90,16%	1,64%	1,64%	6,56%
#8	83	93,98%	0%	4,82%	1,20%
Average	**75**	**89,24%**	**1,48%**	**5,05%**	**4,22%**

Table 4 Transition accuracy during presentations navigated with commands

During the presentations, the indicated familiarity with giving presentations showed itself for most of the speakers. Those that indicated to have some experience with giving presentations, and feeling at ease before a crowd hardly looked at the contents of the slides, other than to keep track of how far they progressed on any given slide. The more inexperienced subjects used the contents of the slides as a clear basis for their speech, literally uttering most of the content elements.

As the results clearly show, uttering commands during a presentation in a quiet environment yields great performance. It does however come across unnatural, as the commands can hardly be embedded into natural speech. Moreover, issuing commands in this way can be considered more intrusive then using a small conceived device to trigger a transition manually.

6.9 Title transition

Starting at slide 1, the presenter progressively transitions to the next slide until the end of the presentation is reached. The experiment setup has been configured in such a way that only the title of the next slide is taken into account.

After explaining the workings of this version of the presentation assistant, the subjects were asked to adjust the presentation to fit their needs. In order to set a base measure, they were asked to read aloud the titles of their presentation. Below are the results of the recognition:

Speaker	#utt.	Correct	Wrong	FalsePos	Missed
#1	16	93,75%	6,25%	0%	0%
#2	22	100%	0%	0%	0%
#3	27	96,30%	0%	3,70%	0%
#4	14	92,86%	0%	7,14%	0%
#5	13	100%	0%	0%	0%
#6	15	100%	0%	0%	0%

#7	25	100%	0%	0%	0%
#8	29	96,55%	0%	3,45%	0%
Average	**20**	**97,43%**	**0,78%**	**1,79%**	**0%**

Table 5 Uttering the titles of the slides

The results are on par with the results of uttering the commands. The utterances show great similarity in the way that silences are present prior and just after uttering a title.

After reading aloud the titles, the participants were asked to give the presentation again, this time allowing them to initialize transitions on the titles as well. The results are displayed in table 6:

Speaker	#utt.	Correct	Wrong	FalsePos	Missed
#1	89	87,64%	1,12%	11,23%	0%
#2	80	92,59%	0%	7,41%	0%
#3	41	70,73%	0%	26,83%	2,44%
#4	38	86,84%	0%	10,53%	2,63%
#5	99	84,85%	0%	15,15%	0%
#6	34	82,35%	0%	11,76%	5,88%
#7	57	80,70%	0%	12,28%	7,02%
#8	58	87,93%	0%	12,07%	0%
Average	**62**	**84,20%**	**0,14%**	**13,41%**	**2,25%**

Table 6 Transition on commands and titles of slides

6.9.1 False positives

This particular experiment aimed at triggering a false positive by offering the subjects a table of contents in which the title of the next slide was already embedded. When uttering that item of the table of contents, the system would respond with triggering a slide transition, after which the subject would hopefully recover by either issuing a command or by continuing their presentation without interruption. This would introduce both a false positive in the recognition, and see how the test subjects would recover from unexpected behaviour of the system.

Although this partially succeeded (two out three presenters regained their composure by issuing a command to return to the table of contents), table 6 shows a vast amount of false positives. This due to three aspects:

1. Some of the titles used in the presentation are single, short words, consisting of only a few phonemes.
2. The desire to display a slide further on in the presentation merely based on uttering the title is unlikely, so including it increases the chance of unwanted behaviour.
3. Using a closed vocabulary FSG recognizer that has single words reach an accepting state results in a large error rate compared to longer sequences.

This is inherent to the designed system. The test subjects are free to adjust the presentation to their liking, and given the typical information denseness found in presentations, short titles are likely to occur in a live situation.

6.9.2 Sidestep: global title transitions

In the previous section we mentioned that presentations are likely to be given in a linear fashion. What would happen if we were to neglect that assumption? Since the audio has been gathered for offline analysis already, we configured a presentation assistant to take all titles into account. A grammar was constructed on each presentation belonging to the previous phase, and the audio was recognized again using this configuration.

The results are shown below in tables 7and 8.

Speaker	#utt.	Correct	Wrong	FalsePos	Missed
#1	16	93,75%	6,25%	0%	0%
#2	22	90,91%	0%	0%	9,09%
#3	27	96,30%	0%	3,70%	0%
#4	14	78,57%	0%	21,43%	0%
#5	13	84,62%	0%	15,38%	0%
#6	15	100%	0%	0%	0%
#7	25	88,00%	0%	12,00%	0%
#8	29	82,76%	0%	17,24%	0%
Average	**20**	**89,36%**	**0,78%**	**8,72%**	**1,14%**

Table 7 Recognition accuracy of uttering titles with a global grammar

The performance of recognizing the correct title when only titles are uttered is still quite good, with an average accuracy of almost 90%. Table 8 shows a different outcome when the speech is the conversational speech during the presentation:

Speaker	#utt.	Correct	Wrong	FalsePos	Missed
#1	89	58,43%	3,37%	38,20%	0%
#2	80	77,50%	0%	22,50%	0%
#3	41	39,02%	9,76%	51,22%	0%
#4	38	73,68%	7,89%	18.42%	0%
#5	99	62,63%	3,03%	34,34%	0%
#6	34	67,65%	5,88%	23,53%	2,94%
#7	57	77,19%	3,51%	17,54%	1,75%
#8	58	70,69%	3,45%	25,86%	0%
Average	**62**	**65,85%**	**4,61%**	**30,46%**	**0,59%**

Table 8 Transition accuracy with the global grammar

As the results show, the amount of raised false positives is staggering: on average, over 30% of the utterances result in a false positive. Almost one in three utterances of the presenter will result in the unwanted behavior of an undesired slide transition with this system.

6.10 Content-based transitions

The last developed presentation assistant still enforces the principle of linear presenting. It includes all content elements of the next slide, including annotations or key phrases put in the *Notes* section of a slide. As described in the approach, this allows for tagging non-textual content.

During the presentation adjusting, only two subjects implemented this in their presentations. Both also uttered the key phrases in order to properly navigate through the presentation.

Speaker	#utt.	Correct	Wrong	FalsePos	Missed
#1	106	99,40%	0%	12,90%	6,60%
#2	36	80,56%	0%	0%	19,44%
#3	63	100%	0%	0%	0%
#4	31	90,32%	0%	0%	9,68%
#5	67	97,01%	0%	2,99%	0%
#6	39	87.18%	0%	0%	12,82%
#7	37	97,30%	0%	2,70%	0%
#8	60	97,67%	0%	0%	2,33%
Average	**55**	**94,61%**	**0%**	**2,32%**	**6,36%**

Table 9 Uttering the content elements of the slides

Even when the grammars become extensive, mapping the speech to the correct content element performs very well. In table 9, the false positives were mostly generated by falsely recognizing something during a silence period, and the missed elements were mostly due to the test subjects uttering the content in another way then they formulated it on the slide itself.

Speaker	#utt.	Correct	Wrong	FalsePos	Missed
#1	93	87,10%	0%	12,90%	0%
#2	86	86,05%	0%	12,79%	1,16%
#3	54	68,52%	0%	29,63%	1,86%
#4	-	-	-	-	-
#5	80	78,75%	0%	20%	1,25%
#6	57	94,74%	0%	5,26%	0%
#7	52	80,77%	0%	11,54%	7,70%
#8	79	83,54%	0%	15,19%	1,27%
Average	**72**	**82,78%**	**0%**	**15,33%**	**1,89%**

Table 10 Transition accuracy with grammars based on all content of the next slide

The audio of speaker 5 for this experiment was so badly clipped (over 85% of the samples), we decided to discard that part for analysis purposes.

The other experiments show a very decent accuracy, on average 83% of all utterances result in the desired behaviour. The amount of false positives is still a bit high, but can be explained again with the fact that most presenters only have the word “Vragen” (questions) on the last slide.

6.11 Interpretation of the results

Given the collected data displayed in the last section, we see a few trends emerging.

6.11.1 Issuing commands

Some of the speakers used the ability to fall back on issuing commands for navigating through the presentation. Whenever they found themselves lost, either due to the unfamiliarity with operating the system, or triggering a transition by accident, it did provide the necessary means to return to the desired slide and continue the presentation. We want to emphasize the importance of the availability of commands as a means to explicitly navigate the presentation when desired; presenters could find themselves lost without them.

6.11.2 Silence periods

The data gained from the experiments shows a great many false positives. Using short words in elements that can trigger a transition, along with the closed vocabulary argument results in a massive amount of transitions, whereas from the speech of the presenter it is clear that it was not an intended one.

The cause of this unwanted behaviour, as discussed in section 6.9.1, is due to the short, often common words. A solution could be to demand (short) silence periods prior and after issuing a command. We do feel that this would void the design goal that the presenter can fluently give his/her presentation. Issuing commands (either direct or by uttering content-related words) with pre- and post periods of silence would result in the presenter having to explicitly navigate the presentation.

A quick scan over the results indicate that although the false positive rate would decrease, some missed transitions would occur as well.

6.12 Strong and weak aspects of the assistants

When analyzing how the content of the slides relate to the performance, some common demeanours appear. These will shortly be discussed in the following paragraphs.

6.12.1 Content length

When constructing grammars based on the next slide, we noticed a significant increase of the false positives near the last slide. This is likely to be caused by the fact that most of the participants had the last slide consist of just 1 word (*vragen*). Only 1 participant placed information on the last slide as well, to which the system properly responded.

Combined with the arguments we presented in section 6.9.1, we can identify the first weakness:

- If the generated rules are based on merely 1 short word, false positives are likely to occur.

The opposite of this resembles a strong aspect of the presentation assistant:

- If a content element consists of a few words, grammar-based recognition is a strong tool to identify this in the speech of the presenter.

However, if this is taken too far, another weak point is introduced:

- If a content element consist of many words (say more then 5), it is unlikely that Spraak will properly identify this part in the speech.

6.12.2 Command confusion

The commands included in every version of the presentation assistant included next() and previous(). In Dutch, these command are (in semi-BNF notation):

```
next = [toon de] volgende [sheet | slide]

prev = [toon de] vorige [sheet | slide]
```

The distinguishing word between the two commands is “volgende” versus “vorige”. Here are their phonetic descriptions:

```
vorige = v o r @ G @

volgende = v o l G @ n d @
```

The distinguishing phonemes are “r” vs “l G” and G vs “ n d”. The difference between the two commands is thus relatively small, explaining the confusion between them during the experiments.

6.12.3 Performance analysis

The proposed linear enforcement yields very decent results. Each of the constructed systems reaches an accuracy of 83% or higher. This number could be even higher if presenters were required to assign more than just one word to a slide, even by using the notes section. This would however impose a restriction on the system, and would thus violate the initial statement that presenters were free to design the slides to their liking.

Although the overall accuracy has been discussed at length, there is one more aspect to investigate. Whenever an utterance appears that should result in an action (either a direct command, or by uttering content of the next slide), the presenter expects the slide transition. The accuracy of the number of missed or wrongly

recognized commands versus the total amount of uttered commands is displayed in table 11. The data is limited to when a presenter is giving a presentation; uttering the list of commands, titles or contents is excluded. It shows an average accuracy of 71,43%. This means that 3 out of 4 intentions during conversational speech were interpreted correctly.

The relatively low performance is mostly due to the inclusion of the global title data, which had a significant amount of wrongly recognized titles. The majority of them originated in the last slide, which typically had the short title 'vragen' (questions).

Speaker	#Issued	Correct	Wrong	Missed
#1	27	85,19%	14,81%	0%
#2	25	80,00%	16,00%	4,00%
#3	24	50,00%	16,67%	33,33%
#4	19	57,89%	26,32%	15,79%
#5	24	83,33%	12,50%	4,17%
#6	15	83,33%	5,56%	11,11%
#7	31	48,39%	9,68%	41,94%
#8	24	83,33%	8,33%	8,33%
Average	**24**	**71,43%**	**13,73%**	**14,83%**

Table 11 Accuracy of the interpretation of utterances that should have resulted in an action during the experiments. Listing the commands, titles or content is excluded; providing a clear view on the expected behaviour from the presenter's point of view.

When the available commands for a transition are limited to uttering a command or title, the presenter is forced to monitor the slides to check if the correct one is displayed. The addition of including all content of the next slide allows for small mistakes to be made, as chances are high another content element of the slide is uttered.

This phenomenon has been observed during at least two experiments with the last system. The presenter did not utter one content element correctly, or the recognizer failed to properly recognize it as such, but as the next element was discussed, the presentation assistant was able to display the correct slide after all, albeit with a delay.

7 Conclusions

Based on the developed assistants and the evaluation we performed, this chapter concludes this thesis by drawing conclusions and offering recommendations for this and future work.

7.1 The assistant

Our approach consisted of several designs, implementations and evaluations of a grammar-based presentation assistant. As the experiments show, the accuracy of the developed systems varied greatly. We observed that enforcing a linear fashion of giving the presentation closely follows the test subjects' expectations, and that including all the content of a slide proved most valuable.

The sidestep with transitioning on each title resulted in a drastic fall of performance. We strongly discourage grammar-based recognition to include content of all slides at this point.

Including all content elements of a next slide allowed the presentation assistant to recover from failing to transition when intended without additional interference of the presenter. We conclude this approach to be the most desirable presentation assistant and are content with its performance.

7.1.1 Limitations

The presentation assistant force two restrictions on the presentations they can be used on:

- The slides cannot contain actions that require user interaction to display elements
- Topic clustering is something that should be avoided if possible, as the assistant is likely to transition too early on extremely similar slides.

In addition, the index content is limited to text. Presenters are allowed and encouraged to annotate non-textual content in order to allow the assistant including these elements.

7.2 The approach

Having designed, developed and evaluated several systems results in gaining insight in the strong and weak point of grammar-based recognition. As presented during the interpretation of the experiment results, we notice that having rules in the grammar that accept just one word leads to a lot of false positives.

The experiments were conducted using a wizard-of-Oz approach, which allowed for a more detailed analysis afterwards, and the availability of fine-tuning the grammars and system. The downside of this approach is that the subjects experienced a flawless system, with the exception of creating a situation where an unwanted slide transition was likely (and happened for 8 out of 9 participants). As a follow-up, it would have been interesting to see the developed system perform in live situations. The absence of a real-time flawless integration with Spraak withheld us from performing this step, but we are confident that additional effort on the integration would provide for a seamless cooperation.

7.2.1 Using Spraak

Although Spraak is one of the most sophisticated and extendable speech recognition systems available, using it an off-the-shelf system in a Windows environment has proven to be a challenge. With the kind help and support of prof. Wambacq, as well as attending a Spraak workshop in December 2008 for my job, using Spraak as a grammar-based recognition system has been realized.

We feel Spraak has more potential than we have been able to explore within this thesis, but having had a closer look at the models has been rewarding. Fine-tuning the finite state grammars and their accompanying

dictionaries has been time-consuming but rewarding, allowing us to improve the performance of the presentation assistant.

For further projects or other thesis implementing speech recognitions, we recommend using Spraak when the need arises for a speech recognition system.

7.2.2 Research goal

If we recall that the original research goal was to investigate if an ASR can determine when a slide transition is in order, we find that answering that question has more nuances to it than a simple yes or no. Basically just a command-driven version of the presentation assistant can meet that goal.

Focusing on the presentation assistant that based its grammars on the entire content of the next slide, we feel that the research goal can be confirmed. When the presenter talks about the next slide, it is most likely that at least one element will be mentioned, which in turn triggers the transition. In some cases, the delay for initiating the transition compared to the speech can be considered too long. If the presenter feels the transition is not initiated when expected, the provided command structure provides an excellent fallback, which was used several times during the experiments.

7.3 Future work

At the start of the thesis, during a brainstorm session, a great variety of ideas were launched. Most of them were not feasible within the time span and effort of a thesis, others showed great potential.

Looking back on the ideas cumulated over the process of designing the assistant and writing the thesis, there are two promising leads that show potential.

7.3.1 Using additional speech and language processing methods

With some of the longer rules in the grammars, we observed that short words from other rules in the grammar are easily inserted in the recognition. We have observed elements like e.g. “aangekomen in Enschede te voet verder” (upon arival in Enschede, continue on foot) to be recognized as “aangekomen de in Enschede de voet te verder”. It is clear that both sentences have the same meaning, yet will not result in a positive match.

The removal of filler words seems to provide a solution for this type of problem, and it does. When in both the content for the rules and the grammar rules stopwords are removed, the above example maps to “aangekomen enschede voet verder” and “aangekomen enschede voet verder” respectively, which is a positive match.

In addition, the addition of compound splitting and stemming could be investigated.

7.3.2 Speech/non speech detection

The number of false positives when the presenter is uttering the commands available for that phase is a lot higher than expected. As a lot of the false positives occurred during turns with no speech, having speech/non speech segmentation on the audio prior to recognizing it might prove very valuable.

Within the limits of this thesis, speech/non speech segmentation has been out of reach, but for instance ELIS (2009) has developed a speech/non-speech (and speaker clustering) that promises great performance. We recommend using reliable speech/non speech segmentation (or at least detection) when using speech recognition systems in a non-controlled environment.

7.3.3 Classification instead of recognition

As this thesis shows, only using the recognition results from an ASR proves error-prone. Additional heuristics provide a safe-warden against unwanted transitions, but they do not cover the base problem: if a recognition contains a keyword, an action is triggered. However, if the recognitions were not directly matched against actions, but rather were used as input for a classification mechanism, the 'hard' triggers of an action would become obsolete.

Using a classification method introduces a new set of challenges. Implementing heuristics such as "show every slide at least 20 seconds" becomes a necessity as most techniques will classify any input (the recognized speech) to an item within the outcome space. This means every classification has the potential to initiate a slide transition.

Instead of using this approach to select the appropriate slide of a presentation, it could become more of an illustration of the presenter's speech. There is no reason why the set of available illustrations should be limited to just presentation slides; any (annotated) content will suffice.

7.3.4 Extending presentation content

During this thesis, the grammars were limited to the actual textual content of the slides. This content was largely provided by the presenter, as text typically makes up for a large portion of any given presentation. With the arrival of embedded videos, audios and the already present graphical images, the need arises to index this type of content as well.

This thesis used the "Notes" section of slides to provide the presenter with the ability to assign keyphrases to slides, or to annotate figures and/or tables contained in a slide. This can be extended with a mechanism to annotate any kind of content.

7.4 Final thoughts

Keyword spotting, closed recognition and dictation systems have become widespread available over the last years. Using speech-enabled car navigation systems, issuing commands to a Smartphone, a great variety of speech-driven IVR phone systems or using a speech-enabled car kit are just basic examples of how speech recognition systems are widely used in everyday life.

The presentation assistant designed, constructed and evaluated in this thesis does however follow the notion of speech-enabled systems in our surroundings. Given the performance of the various versions of the assistant, we are inclined to conclude that a closed system would be preferable, as it enables presenters to include navigation commands (e.g. next slide's title, or a key phrase triggering a transition) into natural speech during a presentation.

This paradigm still holds the notion of direct interaction with the presentation, without the necessity of actively navigating the presentation using a device or other means than conversational speech.

8 Bibliography

Bordegoni, M., Faconti, G., Feiner, S., Maybury, M., Rist, T., Ruggieri, S., et al. (1997). A standard reference model for intelligent multimedia presentation systems. *Computer Standards & Interfaces , 18* (6-7), 477-496.

Chen, S. S., Eide, E. M., Gales, M. J., Gopinath, R. A., Kanevsky, D., & Olsen, P. (1999). Recent improvements to IBM's speech recognition system for automatic transcription of Broadcast News. *IEEE ICASSP-99* , 37-40.

Hain, T., Woodland, P. C., Niesler, T. R., & Whittaker, E. W. (1999). The 1998 HTK system for transcription of conversational telephone speech. *IEEE ICASSP-99* , 57-60.

Halasz, F., & Schwartz, M. The Dexter hypertext reference model. *Communications of the ACM , 37* (2), 30-39.

Huijbregts, M. (2005). *ASR client/server protocol specification.*

Huijbregts, M. (2009). SpraakHerkennings Onderzoek Universiteite Twente (SHOUT). Last viewed on 03-08-2009. http://wwwhome.cs.utwente.nl/~huijbreg/shout/**.**

Faculty Electronics and Informations Systems, University of Gent. (ELIS) (2009). *Audio segmentation*. Last retrieved 08-089-2009 from https://speech.elis.ugent.be/s/index.php?option=content&task=view&id=47.

Department of Electrical Engineering (ESAT); Speech group, Universiteit Leuven (2009). Speech Processing, Recognition & Automatic Annotation Kit (SPRAAK). Last retrieved on 09-08-2009 from http://www.esat.kuleuven.be/psi/spraak/projects/index.php?proj=SPRAAK.

IBM. (2007). *ViaVoice*. Last retrieved on 28-09-2008 from http://www.ibm.com/software/speech

IDIAP Research Institute. (2004). *MLMI04 workshop recordings*. Last viewed on 29-08-2008. http://mmm.idiap.ch/mlmi04/

Jurafsky, D., & Martin, J. H. (2000). *Speech and Language Processing - An introduction to Natural Language Processing, Computational Linguistics and Speech Recognition* (International ed.). Prentice Hall.

Microsoft Corporation. (2007). *Office PowePpoint*. Last retrieved on 28-09-2008. http://office.microsoft.com/en-us/powerpoint/default.aspx

Moscovich, T., Scholz, K., Hughes, J., & Salesin, D. (2004). *Customizable Presentations (Technical Report CS-04-16).* Computer Science Department, Brown University.

Nederlandse Taalunie. Corpus Gesproken Nederlands (CGN). Last retrieved on 16-08-2009 from http://tst.inl.nl/cgndocs/doc_Dutch/start.htm.

Nuance. (2007). *Dragon Naturally Speaking*. Last retrieved on 28-09-2008 from http://www.nuance.com/naturallyspeaking

Nuance. (2008). *Open Speech Recognizer*. Last retrieved on 28-09-2008 from http://www.nuance.com/recognizer/openspeechrecognizer/

Ordelman, R., Huijbregts, M., & de Jong, F. (2005). *Unravelling the Voice of Willem Frederik Hermans: an Oral History Case Study.* University of Twente.

Ordelman, R.J.F. (2009). Twente Nieuws Corpus (TwNC). Startpage last retrieved on 16-08-2009 from http://www.vf.utwente.nl/~druid/TwNC/TwNC-main.html

Ordelman, R.J.F., Jong, F.M.G. de, Huijbregts, M.A.H. & Leeuwen, D.A. van (2005). Robust audio indexing for Dutch Spoken-word Collections. Proceedings fo the XVIth International Conference of the Assocation for History and Computing (AHC2005). KNAW, Amsterdam. ISBN 90-6984045607, pp 215-223.

Pellom, B., & Hacıoğlu, K. (2004). *SONIC: The university of Colorado continuous speech recognizer.* Center for Spoken Language Research, University of Colorado.

Philips. (2007). *Speech Recognition Systems*. Last retrieved on 28-09-2008 from http://www.speechrecognition.philips.com/

Rose, R., & Paul, D. (1990). A hidden Markov model based keyword recognition system. *International conference on acoustics, speech and signal processing, 1990 (ICASSP-90) , 1*, 129-132.

Rutledge, L. a., Ossenbruggen, J. v., Hardman, L., & Geurts, J. (2000). Generating presentation constraints from rhetorical structure. *Proceedings of the 11th ACM on hypertext and hypermedia*, (pp. 19-28).

Infolab, North Western University (2007). *The Intelligent Classroom*. Last retreived on 28-09-2008, from http://infolab.northwestern.edu/project.asp?id=11

STEVIN (2009). Ontwikkelen van een spraakherkenner voor het Nederlands en het aanleggen van een databank voor de semantische verwerking van het Nederlands. Last retrieved on 09-08-2009 from http://taalunieversum.org/taal/technologie/stevin/projecten/#spraak.

University College London, Division of Psychology and Language Sciences. (2009. *Speech Assessment Methods Phonetic Alphabet (SAMPA)*. Last retrieved on 05-08-2009 from http://www.phon.ucl.ac.uk/home/sampa/.

Wagner, W. A., & Fischer, M. J. (1974). The string-to-string correction problem. *Journal of the Assocation for Computing Machinery , 21* (1), 168-173.

Weintraub, M., Taussig, K., Hunicke-Smith, K., & Snodgrass, A. (1996). Effect of speaking style on LVCSR performance. *Proceedings of International Conference on Spoken Language Processing (ICSLP) 1996* , 16-19.

Wilpon, J., DeMarco, D., & Mikkilineni, R. (1988). Isolated word recognition over the DDD telephone network --- results of two extensive field studies. *Proceedings of the IEEE international conference on acoustics, speech and signal processing , 1* (10), 55-58.

Wilpon, J., Miller, L., & Modi, P. (1991). Improvements and applications for key word recognition using hidden Markov models. *IEEE transactions on acoustics, speech and signal processing* , 309-312.

Wilpon, J., Rabiner, L., Lee, C., & Goldman, E. (1990). Automatic recognition of keywords in unconstrained speech using hidden Markov models. *IEEE transactions on acoustics, speech and signal processing , 38* (11), 1870-1878.

Zhang, G., Sun, H., Zheng, F., & Wu, W. (2004). Robust speech recognition directed by extended template matching in dialogue systems. *Proceedings of the 5th World Congress on Intelligent Control and Automation* , 4207-4210.

Table of figures and tables

Tables

Figures

Equations