ON

ADDRESSEE PREDICTION

FOR REMOTE HYBRID MEETING SETTINGS

OR

HOW TO USE MULTIPLE MODALITIES IN PREDICTING WHETHER OR NOT YOU ARE BEING ADDRESSED IN A LIVE HYBRID MEETING ENVIRONMENT

By HARM OP DEN AKKER, BSc.,

Student at the University of Twente, Enschede. To obtain the degree of Master of Science in the Human Media Interaction Program.

Under supervision of:

Dr. Dirk Heylen

Dr. Betsy van Dijk

Ir. Dennis Hofs

Enschede, March 19, 2009

Contents

Chapter 1

Introduction and goal

To answer the question "*Who said what to whom?*" is a key part in understanding what is going on in group conversations. The question consists of three parts: *who* is the source of the message, *what* does the message entail, and *to whom* is the message addressed? This master thesis is about the third part of the question: to whom is the message addressed. More specifically, this work tries to create a method for automatically detecting whether a specific member in a group discussion is the intended addressee of an utterance or not. The general approach in this work is the use and creation of machine classifiers. In our case, the machine classifier functions as a piece of software that is working as an assistant for a specific participant in a meeting. The software can help in telling when the participant that it is operating for is being addressed. The details of this will become clear later on. For now, we look at what addressing is and how many different techniques we, humans, deploy in making sure our message reaches the intended audience. The first chapter continues by looking at why we would want to predict this behaviour automatically, and how other researchers in the field have done so.

But we start with a definition of "addressee". The definition of addressee that we use here is the one coined by Goffman in [1, p.10]. The addressees of an utterance are those particular participants of the conversation "*oriented to by the speaker in a manner to suggest that his words are particularly for them, and that some answer is therefore anticipated from them, more so than from the other ratified participants*". Furthermore, there can be any number of addressees for any particular utterance. It may be addressed to an individual, a group, or maybe even to no one (when talking to oneself).

The difficulty of the task of addressee identification depends heavily on the conversational setting. The work in [2] gives an overview on how the nature of a number of conversational aspects change when considering multiple participants instead of the two-party case. It makes the, perhaps slightly

obvious, but nonetheless important, observation that in the case of two participants, addressee identification is trivial. As an observer, you can safely assume that whatever speaker A is saying is addresssed to speaker B and vica versa. If you are part of the conversation yourself, everything that you didn't say is addressed to you. In other settings, such as the "cocktail party" setting, where multiple participants engage in multiple discussions, the task of addressee identification can be daunting. However, the number of participants alone is not the only factor that can increase the complexity of the task. A courtroom hearing, where the flow of conversation is governed by strict rules, is less complex than the aforementioned cocktail party in terms of determining the addressee of uttered phrases, even though the number of participants could be the same.

The setting in which this research is conducted, is meetings. The number of participants in the meetings that we study is usually 4, although this exact number is not a defining property of the setting of this study, it is in fact, as we shall see later on, an important aspect of our study that the number of participants can be varied. It is however well known that the number of participants in a meeting has a profound effect on the dynamics of the meeting. The more participants, the more the dynamics of the meeting change. With more participants, the chances increase that not everyone is actively participating in all the points on the agenda, and even small discussions in subgroups may occur. A system designed and tested on groups of 4 participants may thus perform very badly in meetings with 20 participants. We assume however, that our research generalizes to 'small' meeting groups. The exact number of participants for which our system would still work is hard to predict. As mentioned before, the difficulty of the task of addressee indentification is also dependent on the manner of organisation of the meeting. Because the meetings that we study are led by a discussion leader, we suspect that our work generalizes to any meeting with perhaps even as much as 8 or 9 participants, as long as the discussion is held in a similarly orderly fashion.

The defining property of the setting is the fact that everyone in the meeting is assumed to know that he or she is participating in that meeting, and that he or she knows that the other participants are in that same meeting. This means that anyone can be addressed by anyone at any moment in time, and the participants are assumed to pay at least some attention to what is going on in the meeting. Compare this to the cocktail party setting, where it is very unlikely that you will be addressed by someone on the other side of the room, standing by a different table, surrounded by different people. In terms of [3], all the participants in the meeting are either *active participants* or *side participants*. These are participants that are either speaking them-

selves, being addressed or actively listening in and taking part in the general conversation. This leaves out two other groups: *bystanders* and *eavesdroppers*. Bystanders are those who are not taking part in the conversation, but whose presence is known by the members of the conversation, while eavesdroppers are those whose presence is not know by the others. When we leave out these last two groups, we come to a definition of addressee identification as posed in [4]: "*the problem of addressee identification amounts to the problem of distinguishing the addressee from the side participants in a conversation*".

1.1 How does addressing work?

Whenever you are engaged in a conversation with more than one coparticipant, you have to know if someone is addressing you individually. There are many different ways how we, humans, decide that an utterance was directed to us. The list below sums up the most important ones:

- You were already engaged in a dialogue with the current speaker. In group discussions, dialogues between two participants emerge and dissolve naturally. Whenever you are engaged in such a dialogue, and the speaker does not explicitely address someone else, you can assume that the next thing your co-participant says is also directed at you.

  1. **Albert:** Did you see the match last night, Eric?
  2. **Eric:** You mean Twente - Schalke'04?
  3. **Albert:** Yeah, did you like it?

  Here, utterance 1 is explicitely addressed to Eric by means of using a name. Utterance 3 is implicitely directed at the same speaker, because it is part of the same dialogue. The addressee is implicitely defined by the context of the conversation.

- The example above also displays a form of 'explicit addressing by name'. Assuming your name is "Eric", and you know that the speaker knows you are Eric, and there is no other Eric currently participating in the conversation, it is safe to assume the question was directed at you. This is a very strong form of explicit addressing, because not only does Eric know the question was addressed to him, everyone else who heard the question knows it wasn't directed at them. Besides the use of name to specifically address someone, Lerner, who has described in detail the numerous methods of addressing used in multi-party conversation in [5], notes that "*If one wants to direct a sequence-initiating action unambiguously to a particular coparticipant, then one can addresss that participant with a personal name or other address term,*

such as a term of endearment ('honey') or a categorical term of address ('coach') that applies uniquely to them on that occasion". If the chosen address term can apply to only one particular individual, it is a very effective method indeed. But people use other means of addressing too.

- Gaze has long been known to play an important part in human-human conversation. Already in 1973, [6] investigated the different functions of gaze and notes among other, more social functions, the use in the synchronization of speech. Indeed, one of its speech related functions is the signalling of addressing. Consider the following example, adapted from [5]:

  1. **Nancy:** You see all these cars coming toward you with their headlights.
  2. **Vivian:** Well thank God there weren't that many.
  3. **Michael:** Remember the guy we saw?
  4. **Nancy:** Eh, haha.

  In this example, Michael's remark could have been directed to anyone, or everyone at once. However, because Michael turns his head towards Nancy at the beginning of line 3, she is the most likely (and intended) addressee of the utterance. Because Nancy sees that Michael is looking at her, she knows the utterance is addressed to her, and because the other participants also see that Michael looks at Nancy, they know that they are not addressed. Michael can be said to address implicitly through focus of attention. Direction of gaze is an important tool to indicate your intended addressee, and it reduces the need for more explicit methods like using a name. When gaze is the only indicator for addressing a certain individual, problems may arise if not all members of the conversation were looking at the speaker. In this case, someone who wasn't looking may think he was being addressed because of the content of the utterance, while seeing the gaze of the speaker could have indicated another intended addressee.

- Another obvious method of addressing is using gestures. Although this may not be used that much in small group discussions, it can be used, in combination with gaze, to point out a specific member of a larger audience.

- Finally, if you were not already engaged in a conversation, you were not addressed by name, the speaker did not point at you, and you did not see the speaker's gaze, you can still **know** that you are being addressed based on the content of the utterance. Take the following example, again adapted from [5]:

1. **Curt:** Well, how was the race last night?
2. **Mike:** [nods]
3. **Curt:** Who won the feature?
4. **Mike:** Al won.
5. **Curt:** Who?
6. **Mike:** Al.
7. **Curt:** Al did?

Mike knows that he went to a car race last night, and he knows that Curt knows this, and he knows that no one else went to any car race. This makes him the obvious candidate to answer Curt's question "how was the race last night?". Lerner calls this the "known-in-common circumstances". In this case of 'implicit addressing through shared knowledge', confusion may arise if not all members of the discussion share the knowledge of you being at the car race.

To sum up, there are five distinct conversational elements that can be used for a speaker to indicate his intended addressee, and by which a hearer can know that he or she is being addressed. These five elements are listed below, including a short example of how they can be used:

The context of the utterance. An answer to a question that was just asked is usually intented for the one who asked the question.

The use of explicit addressing. Using either a name, title or term of endearment you can single out your intended addressee.

The focus of attention, or gaze of the speaker. Looking at someone often implies that you're talking to him or her.

The use of gestures. Pointing out an addressee in a large group.

The known-in-common circumstances. Knowing that you are the only member of an audience that is able to answer a question, based on its content, distinctly selects you as the addressee of that question.

Ideally, when designing automatic addressee detection software, these are the elements that should be used and, in some form, provide the input for the system. Unfortunately it is difficult to gain acces to all of this information. Video camera's and microphone's can record conversations; and speech, gaze and gesture information can be extracted from the data. But the last element of the list, the known-in-common circumstances, is much harder to infer for any system. In this work the focus lies on the extracted speech information (Chapter 5) and the focus of attention, based on a participant's gaze (Chapter 6), leaving out the known-in-common circumstances and gestures.

1.2 Why automatic addressee detection?

So why are we interested in automatically identifying the addressee of spoken utterances? One answer would be, to try to confirm the theories of conversational analysis and social psychology, in works such as [3, 1], and to gain a better understanding of how this particular aspect of human-human communication works. For example, [7] uses addressing information to help in analyzing the social structure and dynamics in spontaneous multi-party interaction. The authors develop theories and models of multi-party social interaction with the aim of helping the design of a number of smart multi-party applications, like archival systems for smart meeting rooms, social network analysis and automated volume control in remote conferencing systems. Although the focus of this particular study lies on the analysis of multiple floors in conversation, they remark that in most of these applications "*... a key concern (albeit one usually left for future work) is the development of machine learning models to recognize 'who is talking to whom'...*". Although the study remains largely in the realm of theory, there are also a number of more practical uses.

In [8], an automatic addressee detection module was developed to aid meeting browsing software. The authors notice that "*...systems based on participants' utterances cannot adequetely convey who the addressee is or her/his response, to the viewers, because only selected speakers are shown.*" To help solve this issue, the authors try to predict the addressee by analyzing head pose. For every utterance they derive features that describe the relative duration and frequency of gaze and estimated eye contact. They use this data to train and test a bayesian machine classifier, resulting in 74% correct addressee estimation in three-person conversation. Knowing who the addressee of an utterance is can then be used to display video images of both speaker and addressee in meeting browsers such as in [9, 10, 11]. These meeting browsers are used to search, using different kind of modalities and techniques, through previously recorded meetings, with the goal of, for example, finding out the reasoning behind a decision that was made in a previously held meeting. In order to accomodate the viewing of meetings in such browsers, a rule-based system for automatic video editing, based on a participant's gaze, has been created in [12]. Where other video editing systems that only look at the current speaker might fail to do so, this proposed method can succesfully convey *1) who is talking to whom* and *2) the hearers' response to speakers*, both of which are extremely crucial to understand the flow of conversation.

The use of addressing information could also improve the selection of images in more unconventional, sophisticated meeting layout tools like [13].

The authors have developed a system that generates comic style, or newspaper style summaries of meetings, based on their transcripts. In the comic layout, stills of the video are extracted and used as the background graphics on which the speech balloons are superimposed. The selection of these images is based on the speaker of a particular utterance from the extractive meeting summary. If it is known who the speaker was addressing his/her speech to, a picture including both speaker and addressee could be selected.

In all of the abovementioned examples of applications that use addressing information; the information is used after the meeting has finished. In other words, processing of the meeting can be done *offline*. There are also scenarios where addressing information is required online, i.e. while a meeting is taking place. The AMIDA User Engagement and Floor Control (UEFC) Demo aims to create a meeting assistent agent that helps participants taking part in a meeting from a remote location using teleconferencing software to be more engaged in that meeting. One of the obvious advantages of joining a meeting from your own desktop through remote meeting software is that you do not need to go somewhere physically. Another advantage is that you can continue your daily work, while keeping half an eye on your screen to keep up with the progress of the meeting, and only actively participate if a topic of your own particular interest is being discussed.

But there is a prominent downside to teleconferencing. When you are sharing a meeting room with your co-participants, you would always have a good sense of what is going on in the meeting, even though the current topic might not be of any real interest to you. If your input would suddenly be needed, or your opinion were to be relevant, you would know this, and you could provide your input. You would, for example, notice that people suddenly stopped talking and are looking at you, even if you where dozing off a little. This is different in a teleconferencing scenario; if you don't pay *continuous partial attention* to your meeting software, there would be difficulty for the participants in the meeting room to reach you. If your teleconferencing software knows when you are being addressed by someone in the meeting room, it could alert you through some visual cue or a sound. The development of this specific application of an automatic addressee detection system is the main motivation for this research project. Chapter 2 explains in full detail the design of the UEFC Demonstrator and the role of the Addressee Detection software in it.

1.3 Examples of addressing systems

Because addressee identification is a trivial task in face-to-face conversation analysis, it has not been the focus of much research in the field of compu-

tational linguistics [4]. This section aims to provide an overview of previous work in the field. In order to give the reader an idea of the many different approaches you can take to tackle the problem of addressee identification, we have chosen three fundamentally different approaches from the literature here. The first ([2]) uses a simple set of rules, [14] focusses on head pose and simple speech features in human-robot interaction, while [15] takes a more classical approach using Bayesian Networks and many multimodal features. The three studies are explained shortly below.

Traum [2] suggests a **rule-based algorithm** for automatic addressee detection, which is used in the *Mission Rehearsal Exercise* (MRE) project:

1. **if** utterance specifies addressee (e.g., a vocative or utterance of just a name when not expecting a short answer or clarification of type person)
   then Addressee = specified addressee
2. **else if** speaker of current utterance is the same as the speaker of the immediately previous utterance
   then Addressee = previous addressee
3. **else if** previous speaker is different from current speaker
   then Addressee = previous speaker
4. **else if** unique other conversational participant
   then Addressee = participant
5. **else** Addressee unknown

The paper does not include a performance analysis in the MRE project, but a thorough analysis of the above algorithm has been done on the AMI corpus in [16]. Considering 6590 dialogue acts, only 1897 (28.8%) are predicted correctly, although this bad performance can be ascribed to the large number of 'Group-addressed' dialogue acts in the Corpus, which is not a possible outcome of the algorithm. When leaving out all Group-addressed dialgoue acts, the algorithm scores 1897 out of 3257 correct (58.2%).

Experiments in [14] focus on determining whether someone addressed a robot or a real person. The first experiments focus on the use of Head Pose as the only feature. The results achieved are around 90% accuracy, which seems very high, but the task is relatively easy with only two possible 'targets' for addressing. A second set of experiments uses features derived from speech, like the inclusion of the word 'robot' or an imperative. Results using MultiLayer Perceptron classifiers amount to an accuracy of 82%, with recall of 65% and precision of 69%. The combination of both visual and

speech approaches resulted in a 92% accuracy of determining the robot as addressee.

The work in [15] describes the creation of Bayesian Networks using a classical machine learning approach. The author uses a variety of different features from different modalities like linguistic- and gaze features, and uses Dynamic Bayesian Networks to model the sequential nature of the task[1]. The goal here is to distinguish who is being addressed by an utterance in a meeting setting. A distinction is made between addressing the Group as a whole, and addressing one of four individuals, identified by their seating position. The results achieved are around 75% accuracy.

These are just a few examples of works on automatic addressee detection, illustrating the fact that there are many different settings and equally many different approaches to the problem. The results vary due to the thoroughness of the research as much as due to the difficulty of the setting. It is therefore hard to compare the results of the different works.

Our own approach can be compared best to the last of the three described studies, that of Natasa Jovanovic ([15]). We will re-use many of the features that the author has defined, but translate them to a setting that better fits our purpose: the online meeting assistant agent within the UEFC Demonstrator. The difference in setting between our work and that of Jovanovic will be explained in detail in Chapter 3.

1.4 Structure of the thesis

This first introductionary chapter should give you, the reader, a general idea of what addressing is, why we need software to automatically detect addressing behaviour and how other researchers in the field have tackled the problem so far. The rest of this thesis deals with our work on an automatic addressing system. It consists of seven major chapters, followed by a summary of results and discussion in Chapter 9.

First, we will give a detailed description of the User Engagement and Floor Control Demonstrator in Chapter 2. This demonstrator has already been mentioned earlier as being the larger framework in which this research takes place, and it is therefore useful to know something about its purpose and design. Because the addressing software that is developed is meant to be deployed within this demonstrator, there are some specific requirements to its setup. Chapter 3 describes the approach that is taken to meet these

[1] i.e. the importance of the sequencing of utterances in communication and its ordering.

requirements and explains why it makes this study fundamentally different from other work in the field. Chapter 4 gives a description of the corpus that we've studied and which we use for our machine learning experiments. It also contains an analysis of the reliability of the data by means of looking at inter-annotator agreement.

The next two chapters deal with the core of our work: the development of machine learners for automatic addressee detection. Chapter 5 describes the development of a machine classifier based on linguistic- and context based features: those features that can be derived from the words and dialogue acts in the corpus. Then, Chapter 6 deals with the creation of a classifier that uses Visual Focus of Attention based features, features based on the information of what the participants in the meeting are looking at.

In Chapter 7, we try out three different ways of combining the linguistic- and visual classifiers that were described in the previous two chapters. In the last chapter (Chapter 8), we try to enhance the results of Chapter 7. We try to find out whether knowing the role of the current participant, and the topic that is being discussed can help in the prediction of the addressee of an utterance. Finally, the findings of the research are layed out, and its implications are discussed in Chapter 9.

Chapter 2

The UEFC demonstrator

The automatic addressee detection software that is developed within this research is aimed to be incorporated into the UEFC (or User Engagement and Floor Control) Demonstrator. This tool is a showcase demonstration of technology developed within the European research projects AMI[1] and AMIDA[2]. It has to demonstrate the use of various software components that have been developed over the years by the AMIDA partners. The focus of AMIDA lies on so-called hybrid meetings: those where some of the participants are seated in a common meeting room, and some have joined the meeting through remote communication. So, the demonstrator must focus on this kind of interaction, commonly known as *teleconferencing*.

Conversation between people who are not physically at the same location, such as in a remote teleconferencing meeting, is different, and more difficult, than local conversations in a number of ways. In terms of [3], there are three features of face-to-face communication that do not hold for remote communication and cause problems: a lack of *visibility*, lack of *audibility* and lack of *instantaneity*. Absence of full visibility can cause problems for *addressing* (or *referring* in general [17]), *turn-taking* and *grounding*. The use of a digital audio stream instead of face-to-face talk, is a cause for reduced *audibility* due to microphone glitches, audio feedback or background noise, while the network delay causes a lack of *instantaneity* (see also [18]). These problems eventually cause delay in the speed and quality of conversation with remote participants as well as a reduction in task performance [19]. Another difficulty in mediated communication is that remote participants have a reduced ability to spontaneously take the conversational floor [20]. All this has the effect that remote participants in a meeting feel less engaged in the meeting, which could cause them to lose interest altogether.

[1] Augmented Multiparty Interaction
[2] Augmented Multiparty Interaction with Distance Acces

This then becomes the goal of the UEFC Demonstrator: **to help remote participants to be more engaged, and to facilitate floor control in hybrid meetings**.

One of the envisioned ways to achieve more engagement and facilitate floor control from the side of the remote participant, and thus one of the subgoals of the UEFC Demo Project, is to provide an automated way of notifying the remote participant when he or she is being addressed. On the one side, this will hopefully have the effect of speeding up the conversation with the remote participant who is trying to pay attention to the meeting, but sometimes fails to understand everything that is going on. In other words: it can help in changing the floor between local and remote participants quickly. On the other side it can facilitate a multi-tasking remote participant, who does not try to follow the entire meeting at all. This scenario of continuous partial attention [21] is a frequently occuring one, where participants join a meeting remotely, even though travel time can be neglected, because only certain agenda items concerns them. These type of participants will generally keep the meeting software running, and continue working on something else during the meeting. If their attention is then required, because a specific agenda item has come up, or their opinion is requested, it would be useful to have the system automatically notify the remote participant. In this way, remote participants can be more engaged, by making sure they don't miss out information that they are interested in.

In order to test the software for automated assistance to the remote participant in a meeting, two things are needed: an instrumented meeting room and the software to connect to the meeting room remotely. The next two sections give a detailed overview of the workings of the UEFC Demonstrator. Section 2.1 describes the setup from a hardware perspective, while Section 2.2 gives more details on overall software architecture and the various components.

2.1 Hardware and meeting room layout

The UEFC Demonstrator consists of a video- and audio instrumented meeting room, currently residing in the Computer Science building of the University of Twente, and a video-conferencing tool that can be run from any location on any PC or laptop.

Figure 2.1 shows a schematic representation of the meeting room and remote participant location used for the UEFC Demo.

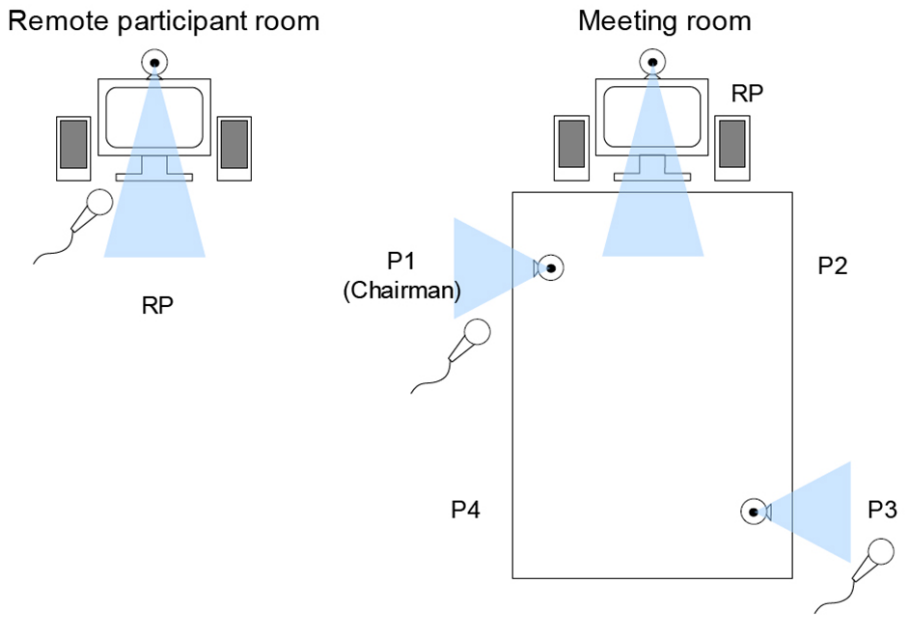


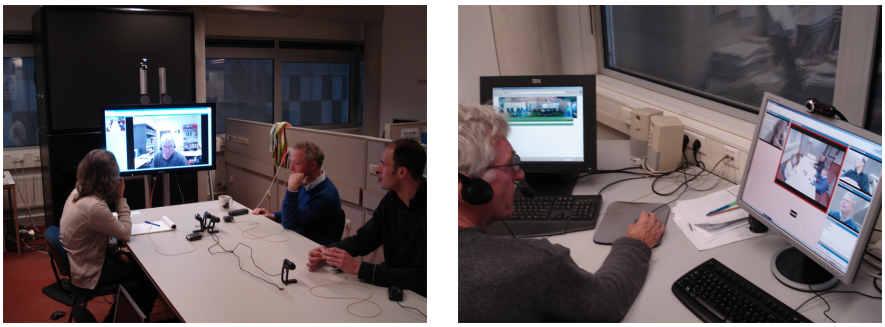
Figure 2.1: Schematic Representation of the UEFC Demo Meeting Room Layout

The meeting room is currently set up for a maximum of four local participants, $P_1..P_4$. Every participant has a regular webcam in front of him that captures his face and upper body. These webcams are connected to standard PCs, running Windows. The screen at the head of the table is a big screen, with a wide-view webcam on top and two speakers to the side, all connected to a simple PC, also running Windows. For all four local participants their are wireless directional microphones that pick up very little background noise; these are also connected to their respective participant's computers.

On the side of the Remote Participant (RP), any type of computer can be used, as long as it has a microphone, a webcam and speakers/headphones. The Remote Participant can see the video streams from all five cameras in the meeting room, and can hear the four microphone streams, mixed together to a single channel.

Participant 1, called the Chairman, is currently the only participant that has a second webcam in front of him. This webcam is connected to a separate machine which extracts Visual Focus Of Attention (VFOA) information. This system keep tracks of where the participant is looking at at any time.

In the future, every participant's webcam will be used for calculating the VFOA information as well as sending the video to the Remote Participant, but this is a technical difficulty that is not solved at the time of writing.



(a) (b)

Figure 2.2: A live remote meeting in action, with the Instrumented Meeting Room (a) and the Remote Participant who is giving the UEFC sofware partial attention (b).

2.2 Software and architecture

The architecture of the UEFC Demo is based around a number of seperate modules, listed below. The details of these will be explained later on.

- The Media Streamer handles the video and audio streams between participants (2.2.1).
- The Hub is a central database for distributing information between participants (2.2.2).
- The Automatic Speech Recognizer (ASR) transforms speech to text (2.2.3).
- The Dialogue Act Recognizer (DAR) segments text into dialogue acts and labels them (2.2.4).
- The Keyword Spotter (KWS) can signal with high precision when certain words are uttered (2.2.5).
- The Visual Focus of Attention (VFOA) module keeps track of who/what the participants are looking at (2.2.6).
- The Automatic Addressee Detection module is the software that is developed in this Thesis (2.2.7).

Some of these modules receive a direct audio or video stream as input. Figure 2.3 describes the audio streams in the setup. The arrowheads indicate the direction of the data stream. The Remote Participant client sends its audio to the Meeting Room PC and to the Automatic Speech Recognition and

KeyWordSpotter modules. The local participant Clients A,B,C and D send their data to the Remote Participant and to the ASR and KWS modules.

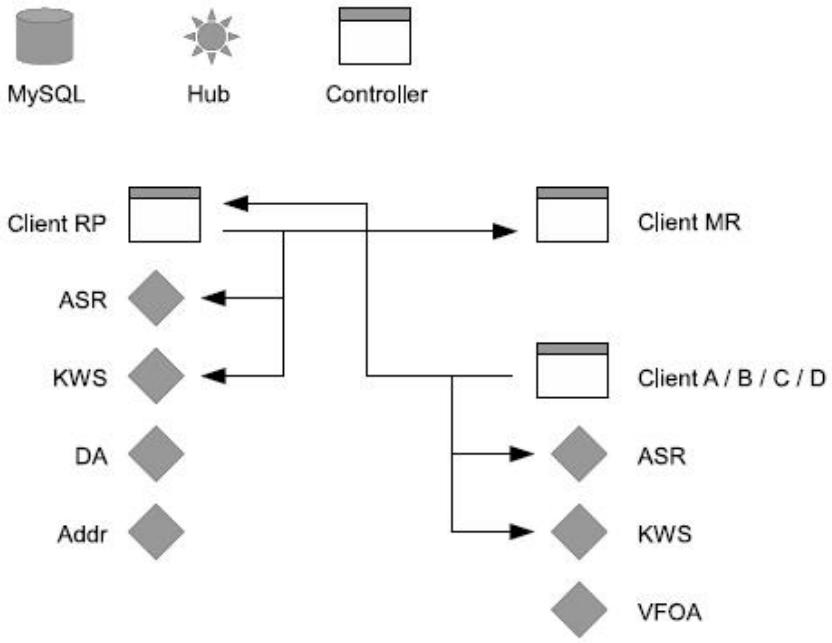


Figure 2.3: Data flow diagram of all audio streams within the UEFC Architecture.

The diagram in Figure 2.4 shows the flow of video data streams. The Remote Participant and Meeting Room clients send their video through to each other. The local Clients A,B,C and D send their video data to the Remote Participant as well as to the Visual Focus of Attention Module.

The three modules (ASR, VFOA and KWS) that receive media input streams, send their respective outputs to a central Database application known as *The Hub*, which sends it through to the modules that rely on the data. Figure 2.5 shows the dependencies of all modules between each other. The Media Streamer records all video and audio from the local and remote participants; the ASR, VFOA and KWS process video or audio data, which is used by the Dialogue Act Recognizer and the Addressing Module. The sections below will explain the details of all the individual modules in the system.

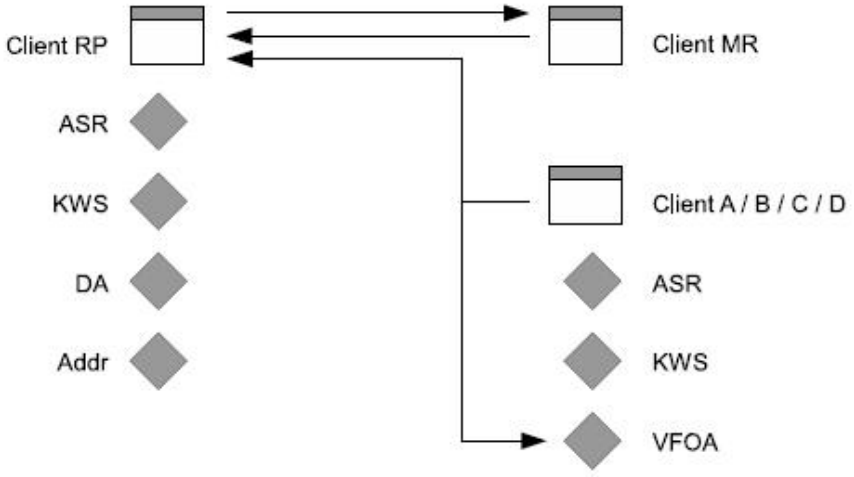


Figure 2.4: Data flow diagram of the video streams within the UEFC Architecture.

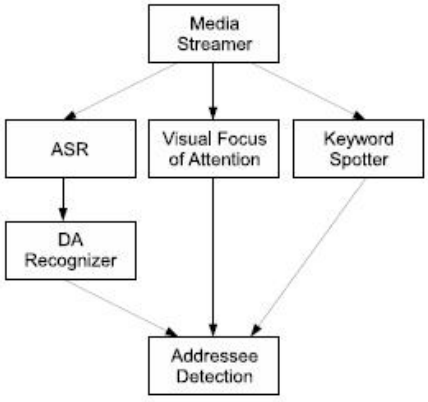


Figure 2.5: Dependencies between all individual modules within the UEFC Architecture.

2.2.1 Media streamer

The Media Streamer is a videoconferencing tool developed at the University of Twente. It runs on all participant's computers and that of the meeting

room itself. It reads out the data from the webcam and microphone, and can stream the data to another PC for further processing. It also takes care of compressing the video stream. The audio and data streams can be seen in Figures 2.3 and 2.4 respectively.

2.2.2 The hub

The Hub is originally developed for the AMIDA Content Linking Device. This is the other major AMIDA Demonstrator application that aims to retrieve documents relevant to an ongoing meeting on the fly [22]. The Hub serves as a central point of communication for different software modules developed within AMIDA. Modules can *subscribe* to the Hub as a producer or consumer (or both) of specific types of data. In our UEFC example, the Visual Focus of Attention Module produces "focus" data, which is consumed by the Addressing module, which in its turn produces "addressing" data. The Hub makes sure that every module is aware of new data arriving from other modules.

2.2.3 Automatic speech recognition

The ASR systems receives the incoming audio streams from all participants on different sockets. For every audio stream it generates the words that are being spoken. It does this in spurts; there needs to be a short silence before the system starts to process the stream. The word data is send to the Hub, including start- and end time information, and from which of the participants it came. The system that is used within the UEFC Demonstrator is the webASR system from the University of Sheffield. For the details on this system please refer to [23][3].

2.2.4 Dialogue act recognition

The Dialogue Act Recognition module segments the words from the ASR module into Dialogue Act segments and classifies them with a Dialogue Act Tag from the AMI tag set (see Chapter 5). At the time of writing the segmentation is done using so-called *spurts*, meaning that a segment boundary is inserted whenever there is a pause of a certain size between two words. In the future, the segmentation algorithm described in [24] will be used. The Dialogue Act Classification, or tagging, is done using the system described in [25].

2.2.5 Keyword spotting

The Keyword Spotting module analyses the audio input stream for the occurence of certain keywords. It can be given a list of keywords, that can be

[3] webASR is located at the following website: http://webasr.dcs.shef.ac.uk/.

modified on the fly, for which it will look. Whenever it detects on of the keywords, it sends a signal to the hub, indicating the word and the time in the audio stream at which it recognized it. The module can handle a list of up to 100 words, and is, for these words, much more reliable than the standard ASR system. The keywords for which spotter looks are inserted by the Remote Participant, so that he can be warned whenever a topic of his interest is being discussed.

2.2.6 Visual focus of attention recognition

The Visual Focus of Attention module analyses the video streams of each individual meeting participant. It tracks the pose of the head in terms of tilt (vertical movement) and pan (horizontal movement) and maps these values to predefined targets. The system then sends for every 2 frames of video data (e.g. 15 times per second) the best matching target to the Hub. In the current setup we are only interested in who is looking at the Remote Participant's screen, so there are two targets: *remote participant* and *other*. The system that is used is based on work in [26].

2.2.7 Automatic addressee detection

The Automatic Addressee Detection module runs for the remote participant only. It receives the data from the Dialogue Act Recognizer, the Visual Focus of Attention Module and the Keyword Spotting, and has to determine whether the remote participant is being addressed or not. If it detects that a Dialogue Act is addressed to the remote participant for which is it running, it will send a signal back to the Hub, which can be picked up by the Remote Participant's interface (see Figure 2.6). This report describes in detail the design of this module.

2.3 Interface prototype

The functionality of all the combined modules is aimed at providing the remote participant with tools that can help him to be more engaged, and more easily obtain the floor in the meeting. Therefore, a prototype interface is developed in which all the functionality is built in. Figure 2.6 shows a screenshot of this prototype. It shows the overview of the meeting room in the middle, which is currently "lit up" by the red border, indicating that the user is being addressed. The transcript window shows the output of the ASR module. The words that are marked red have been spotted by the Keyword Spotter. In the window "Keywords to spot" in the lower right, the list of keywords that the user is interested in can be modified. Below that you can indicate your status as "attentive" (disable all warnings) and "alert me" (warn me whenever I am addressed, or a keyword is spotted).

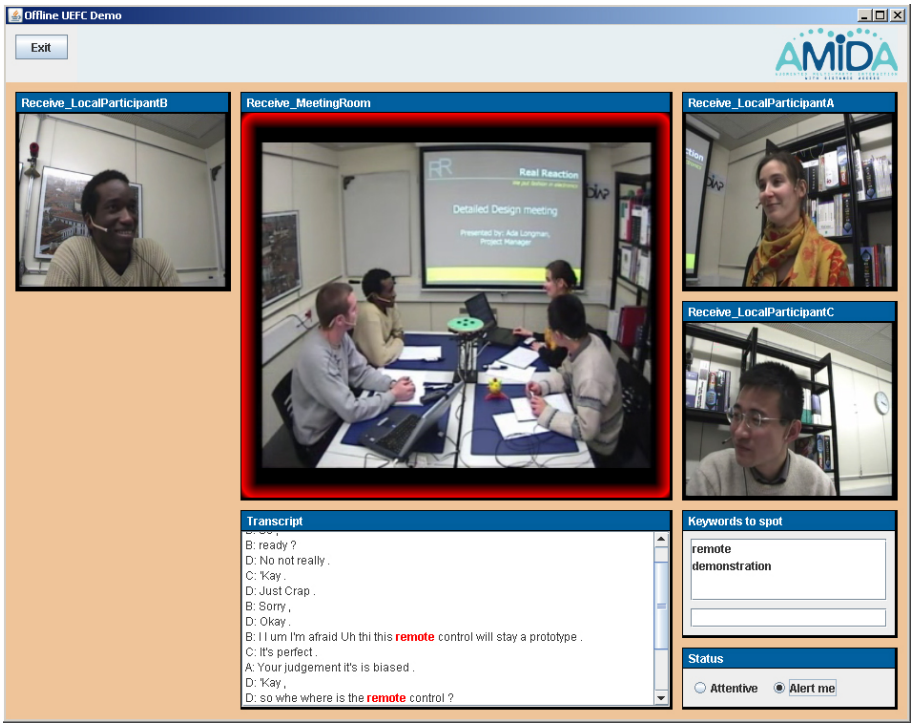


Figure 2.6: Screenshot of the Remote Participant interface of the UEFC Demo.

This Chapter hopefully gave an idea of the functionality of the User Engagement and Floor Control Demonstrator. It is not very important to fully understand the architecture and all the individual modules in detail. More importantly, you should have an idea how you could use the software, sitting

behind your desk, and having in front of you a program like the one depicted in Figure 2.6, and what the role of the addressee detection module is within it. The next Chapter will explain how the problem of addressee detection should be approached in order for it to work within this demonstrator software.

Chapter 3

Addressee classification setting

Most research in the field of automatic addressee detection tries to answer the question of "who *talked* to whom" [4]. Although this may seem to be an accurate desciption of the problem statement, there is a problem of semantics. Perhaps involuntarily, the question seems to be expanded to something like *for all the participants in a conversation, are his or her utterances directed towards: a) the group as a whole, b) a subgroup of participants, or c) a specific participant* [27]. It would not be a problem, per se, to make a system that can classify utterances according to those three categories, although the distinction between the entire group and a subgroup of participants may prove to be hard to make. But the real problem is that we are not so much interested in whether or not an individual is being addressed, we would rather like to know who exactly is being addressed. In order for an automated system to do that (naming a specific person as being the addressee of an utterance), the participants of a conversation need to be identified.

This is the point where assumptions typically have to be made. It should, for example, be known what the possible options for addressee targets. Or, you need to know who are currently participating in the discussion and how each indiviual is identified. In [27], the following assumption for the addressee prediction algorithm is made. *Given that each meeting in the corpus consists of four participants, the addressee tag set contains the following values*:

- a single participant: P_x
- a subgroup of participants: P_x,P_y
- the whole audience: P_x,P_y,P_z
- *Unknown*

Furthermore, the participants ($P_x, x = 0, 1, 2, 3$) are identified by their seating position around a square table (see Figure 3.1).

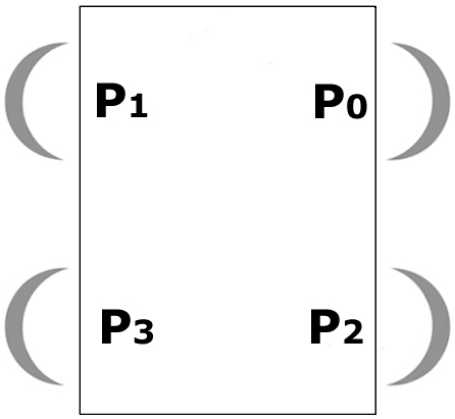


Figure 3.1: Fixed seating positions around a square table.

These two assumptions may or may not be acceptable for the application of the research, and in this case where no specific application is foreseen, they certainly are. However, they are still assumptions that do not always hold.

Another approach to the question of who addressed whom, regarding the identification of discussion participants and the number of participants in that discussion can be seen in [2]. Here, no assumptions are specifically made on the number of participants and the way they are identified. Addressees are simply, and rather vaguely, identified by name or 'a vocative'. This of course implicitely assumes an intelligence in the system that can link names, nicknames or indications of social status to objects that can be identified to be specific persons. Unfortunately, the assumption that this sort of technology exists is not a realistic one, and the issue remains unsolved for a realistic application of automatic addressee detection software.

For research purposes however, assuming a fixed number of participants on fixed positions around a table is reasonable, and important progress have been booked in the field by these studies. For this study, the assumptions are too strict. The automatic addressee detection system that will be developed in this Master Thesis will be incorporated into the previously mentioned AMIDA User Engagement and Floor Control Demonstrator (Chapter 2). Therefore, the requirements for the addressee detection system are a bit more demanding. The following three requirements are the most important:

1. Must accomodate for a remote participant.
2. Should work with variable numbers of meeting participants.
3. Must work in real-time.

These requirements call for a different approach to the problem of addressee detection, that does not have the assumptions of [15] or [2]. Because the remote participant is the center of attention for the UEFC Demo, the focus for addressing in this work will also lie here. Instead of trying to keep track who is addressing whom in the entire meeting, we take in the position of a "support agent" that works for the remote participant and is able to warn him when he is being addressed. This translates the problem statement from "who is addressing whom?" to:

"*am I being addressed?*"

With this question, there is no need to personally identify any of the other participants in the meeting, and there is no need to know where everyone else is sitting (or standing/walking). Therefore, there is no restriction in terms of the amount of participants in the meeting, and it is not required to know this amount beforehand. Intuitively, the task also becomes easier, because you do not need to keep track of who is talking to whom for all the participants in the meeting, you only care about "yourself" (or the person for which the software is running). And if other participants also need to know when they are being addressed, they can use the same system, which will keep track of things for *them*. Please note that our definition of "addressed to me" does not include utterances that are addressed to a group of individuals where *I* am part of, but only those that are *specifically addressed to me and to no one else*.

This different addressee classification setting handles the first two requirements mentioned above, but not the third one. The online (or: 'real-time') requirement limits the input data for the system to past information. So, when analysing an utterance, to find out if it is addressed to "me"[1], we cannot wait for response from other participants, or see where everyone is looking at after the statement. This is a huge disadvantage compared to addressing systems that work after the meeting is done (e.g. for use in meeting browsers). For example, when analyzing the following conversation:

1. Participant A: What do you think about that?
2. Participant B: I agree completely.

[1]From now on in this report, the 'me'-person is the individual in the meeting for which our software agent is trying to determine whether he or she is being addressed.

Giving the fact that Participant B speaks after sentence 1, the probability that sentence 1 was addressed to Participant B is obviously much higher[2]. An online system has to determine the addressee of sentence 1 without knowing who will answer this question.

To conclude; the automatic addressee detection module developed within this work distinguishes itself on two fronts: 1) it works as an assisting agent for one particular individual, and 2) it works in a real-time setting. Now that we have determined our approach in the problem of addressee detection, it is time to solve it. But before we get to the creation of the machine classifiers, the next Chapter will first explain the details of the data that we use.

[2]Although it is still not certain that sentence 1 was indeed addressed to Participant B, he/she may be mistaken as well.

Chapter 4

The AMI corpus

"*Machine learning is programming computers to optimize a performance criterion using example data or past experience*". This is the definition of machine learning given by [28]. Put into terms of addressing, we look at example utterances of people of which we know who they are addressed to, and try to infer rules or generalizations from that data in order to predict the addressee of unseen examples. So we need examples: utterances in a multiparty conversation that have been annotated by humans, so that we know who they are addressed to. Luckily, this sort of information is available. The data used in this project comes from the AMI corpus [29]. This corpus is a huge collection of recorded and hand-annotated meetings which was created for the purpose of analyzing group conversational behaviour within the AMI Consortium[1]. Figure 4.1 shows some snapshots of the video recording of one of the AMI meetings.

Figure 4.1: Snapshots of a recording of a typical AMI Meeting.

[1] http://www.amiproject.org/

In order to be able to analyze addressing behaviour in these meetings, they have to include annotations of which parts of the speech where addressed to whom. This has been done in the following way. Every meeting has four participants, named *Participant A* through *Participant D*. For each of these participants their speech has been hand-annotated (time aligned to the video/audio tracks) in the word-layer. These hand annotated words have then been segmented into Dialogue Act units. A Dialogue Act is defined as a sequence of subsequent words from a single speaker that form a single statement, an intention or an expression. The addressing annotation is thus done on the Dialogue Act level: every dialogue act has one addressee label. This can either be one of the other participants: A, B, C or D[2] or the Group in its entirety.

Besides the word, dialogue acts, and addressing annotations, there are three other layers of annotation that are used in this research. The exact usage of them will become clear in future chapters, but they are listed here for completeness:

- Role. For every participant, their role in the meeting as either *Project Manager*, *Marketing Expert*, *Industrial Designer* or *User Interface Specialist* is documented.
- Topic. Every discussion is divided into broad topics like '*opening*' or '*interface specialist presentation*'.
- Focus. For every participant it is documented where their visual focus of attention lies, or, what they are looking at. This can be, for example, '*table*', '*participant A*' or '*unspecified*'.

4.1 Reliability of data

The task of automatic addressee detection is a fairly high level one, in that it relies on the availability of a number of different layers of information. On each of these layers, *automatic* detection can fail, or mistakes in human made annotations can be made. Errors on a low level, mitigate through to the higher levels of analysis, potentially making the high level task impossible to do.

Take for example, a sentence like "Hey Richard, why tell a fan to get lost?", which is a question addressed to Richard. This can change into something completely different in the process of automatic analysis:

ASR Errors: hey pitcher white elephant who get lost

[2]Note that a Dialogue Act is never addressed to the speaker of that Dialogue Act.

False Dialogue Act Segments: hey pitcher — white elephant — who get lost

False Dialogue Act Labels: Be-Positive — Statement — Elicit-Inform

After this, there is no chance for an Addressing System to do its task right. This error starts at the level of the Automatic Speech Recognition, throwing Dialogue Act Segmentation and Classification completely off course, but errors could also start at these higher levels. Worse still, even if speech detection and dialogue act recognizers work perfectly, an addressee detection system can be deceived by erroneous focus of attention information. This error mitigation phenomena could (and probably will) cause major problems in a fully online system, such as the User Engagement and Floor Control Demo.

But for now, the outlook is less grim. For training and testing our classifiers, we only use the hand annotated (or gold standard) data. The quality of this data is still far superior to that which online systems can achieve, especially on a fairly unambiguous task like transcribing speech. It can happen that an annotator doing speech transcriptions misheard someone because of bad audio quality or a mumbling participant; but the task of writing down what someone is saying is not *vague* or *ambiguous* in essence. The same can not be said for dialogue act segmentation, dialogue act classification, visual focus of attention annotation or the applying of addressee labels. These tasks all require a level of human judgement: it is not always clear where a dialogue act ends or starts, it is not always clear what label it should have, and it is not always clear who the addressee of a particular utterance is (otherwise we wouldn't need to study the problem).

We have reason to assume however, that the data is reliable enough to work with. [15] presents pairwise annotator agreement figures on the Visual Focus of Attention data with good Kappa[3] values, ranging between 0.84 and 0.95, which is an indication of very good agreement. The work in [24] reports an average F-Measure of 0.85 for inter-annotator comparison in the Dialogue Act Segmentation task, with Precision and Recall values ranging between 0.72 and 0.94. This is also a very good score, especially considering the fact that many mistakes can be classified as *harmless* [24].

For the addressee annotations we are interested in a more detailed analysis of the inter annotator agreement. One of the AMI meetings[4] has been

[3]The Kappa Cohen value is a reliability metric that normalizes the agreement percentage of a pair of annotators with the expected agreement by chance, see [30].

[4]IS1003d

annotated with addressing information by four different annotators. We will use this to see how much agreement there is on the data, and use this as a measure of how ambiguous the task of addressee labeling is. Table 4.1 shows the confusion matrix for two annotators: *s9553330* and *vkaraisk*. This gives an idea of the amount of agreement for labelling dialogue acts as addressed to speaker A, B, C, D or the Group. However, because we use our data differently (*am I being addressed?*), we need to look at the confusion matrices in a different way. We can split it up into 4 matrices, each from the view of one of the four meeting participants. Table 4.2 is an example of this, taking the view of participant A, and having annotator s9553330 as gold standard.

	A	B	C	D	Group	Total
A	**29**				10	39
B		**14**			8	22
C			**32**		7	39
D	1		1	**49**	18	69
Group	21	10	19	22	**171**	243
Total	51	24	52	71	214	412

Table 4.1: Confusion matrix for pair s9553330 and vkaraisk. Alpha Krippendorff: 0.55, Kappa Cohen: 0.55.

	A	$\neg A$	Total
A	**29**	10	39
$\neg A$	22	**351**	373
Total	51	361	412

Table 4.2: Confusion matrix for pair s955330 and vkaraisk, considering addressed to A or not.

Table 4.2 shows that when taking annotator s9553330 as gold standard, and considering annotator vkaraisk as the classifier, he achieves an accuracy of 92,23% (380 out of 412 instances classified correctly). When we look at these human annotators as classifiers, we can use their scores as a measure of "maximum performance", because it indicates a certain level of task ambiguity.

There is some debate on the statement that inter-annotator agreement measures can serve as a maximum achievable result for classifiers [31]. It is said that classifiers can achieve higher scores, because they can learn through noise in the data. This is true; this inter-annotator confusion value is not an absolute limit of actual performance, but cases in which the classifier is right and the test-set wrong would not be reflected in the results. The

inter annotator confusion does also say something about the inherent task ambiguity, and can therefore be used perfectly well as a measure to compare your classifier score with.

Table 4.3 contains the overal scores (taken over all 4 individual participants) for the 6 annotator pairs. A complete overview of all inter annotator confusion data can be found in Appendix A.

Annotator 1	**Annotator2**	Recall	Precision	F-Measure	Accuracy
s9553330	vkaraisk	73,37	62,63	67,58	92,78
marisa	s9553330	59,75	70,59	64,72	91,87
marisa	vkaraisk	69,92	74,78	72,27	93,11
marisa	dharshi	37,77	81,61	51,64	91,79
vkaraisk	dharshi	42,04	80,49	55,23	92,22
s9553330	dharshi	43,68	77,55	55,88	93,02
	Average:	**54,42**	**74,61**	**61,22**	**92,47**

Table 4.3: Confusion matrix for pair s955330 and vkaraisk, considering addressed to A or not.

The average values for Recall, Precision, F-Measure and Accuracy will be used as a roof to compare the classifier results with in later chapters.

4.2 Train- and test set split

Unfortunately not all meetings in the AMI Corpus are annoted with addressing information, therefore most of the corpus cannot be used in this research area. The meetings that have been annotated with addressing information are split into training- and test set in the same manner as in [15]. A total of 14 meetings are in the training set, and 4 meetings are in the test set. Tables 4.4 and 4.5 show the training- and test meetings respectively. The meetings that have a ✓in the second column also contain *Focus of Attention* information. In these meetings it is annotated where every participant is looking at, at any time. The last column shows the total number of Dialogue Acts uttered in that meeting by all participants.

The data described here will be used in the next two chapters for the training and testing of our machine classifiers. First we will use the word- and dialogue act layer information to create the Linguistic- and Context Based Classifier in Chapter 5, then the Focus of Attention layer is used in Chapter 6 for the Visual Focus of Attention Classifier.

Table 4.4: Training Data Used.

Meeting	**VFOA**	**# DA's**
ES2009c		904
ES2009d		1249
IS1000a	✓	658
IS1001a	✓	323
IS1001b	✓	897
IS1001c	✓	565
IS1006b	✓	953
IS1006d	✓	1232
IS1008a	✓	263
IS1008b	✓	640
IS1008c	✓	584
IS1008d	✓	589
TS3005a	✓	641
TS3005b		1401
Totals:	7345	10899

Table 4.5: Test Data Used.

Meeting	**VFOA**	**# DA's**
ES2008a	✓	386
ES2008b		955
IS1003b	✓	693
IS1003d	✓	1234
Totals:	2313	3268

Chapter 5

The linguistic- and context based classifier

For the User Engagement and Floor Control Demonstrator (see Chapter 2), availability of online visual focus of attention information was not guaranteed at the time of writing. Therefore we decided to build a classifier that uses only features based on the word- and dialogue act layers of annotation. We call these *Linguistic Features*. Although the AMI corpus is large enough to train such a statistical classifier, it is known beforehand that its performance will be unsufficient for a real world application for two reasons:

1. Visual focus of attention seems to be the richest feature for determining addressee's of dialogue acts. Without this, very good results are not expected.

2. In the remote setting, explicit addressing of the remote participant by using names or raising one's voice, could provide valuable features that can not be exploited here. The reason for this is that our data comes from local meetings (without a remote participant), in which explicit addressing by name occurs very rarely.

The assumption that this classifier can still be a useful component of the final system is that most of the language use, and the cues for automatically determining addressee information therein, is largely the same in a local setting compared to the remote setting. For example: "*What is your opinion on this?*" is a type of utterance that can be expected to occur in both settings, and it contains useful information like 1) it is a question, and 2) it is likely to be addressed to an individual (*your*).

The setup of the classifier is different from the one used in [15] as explained in Chapter 3: our addressing module needs to distinguish between *addressed to me* or not, whether the work in [15] focuses on determining

whether an utterance is addressed to one of four particular individuals or the group as a whole. However, the linguistic features described there might still be useful. The information within these features that apparently help distinguish one Dialogue Act as being addressed to indivual A, and another as being addressed to B is still useful in our addressing setting. Therefore, all these features will be implemented for our classifier. A second type of feature that is used here is the context features. These contain information about the current state of the conversation and the state of participation of the user of the system.

Whether or not the classifier can actually use the features that we provide as input can not be known beforehand. Some features might contain useful information and so improve the ability to correctly classify unseen examples, whereas other features might contain too much false information due to annotation errors, or contain only information that is irrelevant for predicting the addressee. Likewise, some features may be weak individually, but could be very valuable in combination with other features. In the evaluation, later in this chapter, we will try out all possible combinations of features in order to find the the optimal feature set. For now, we will describe all the features for which we think they might be valuable. The list of features for the linguistic- and context classifier is as follows:

5.1 Linguistic features

The following sections describe all the features that can be derived from the word and dialogue act level information. The features are derived from [15]. It may not always be clear why a certain feature would help in detecting whether a dialogue act is addressed to me, but the goal here is to define as many features as possible. If they turn out to be useless, they will be filtered out in the feature selection phase in Section 5.3.

Table 5.1 describes how the AMI dialogue act tagset is mapped to a slightly smaller set as defined by [15] to improve information density. The left column contains the dialogue act tagset as used in the AMI corpus, the right column contains the ones used for the features below. A question mark means the da-type will be set to *Missing*.

5.1.1 Type of the current dialogue act

The type of the current dialogue act as determined by the mapping in Table 5.1. If the dialogue act type is either *Backchannel*, Stall or *Fragment*, the dialogue act is never addressed to anyone, as a rule for the annotators.

Table 5.1: Dialogue Act Type mapping table.

AMI Tag	Feature Tag
Backchannel	Backchannel
Stall	Stall
Fragment	Fragment
Inform	Inform
Suggest	Suggest
Assess	Assess
Elicit-Inform	Elicit
Elicit-Offer-Or-Suggestion	Elicit
Elicit-Assessment	Elicit
Elicit-Comment-Understanding	Elicit
Offer	Offer
Comment-About-Understanding	Comment-About-Understanding
Be-Positive	Social
Be-Negative	Social
Other	?
Unlab	?

Therefore, these dialogue acts are not used for training or testing the classifier (as the outcome should always be '**no**, not addressed to you'). However, they are used for determining the contextual features (see Section 5.2).

5.1.2 Short dialogue act

This feature has the value 'yes' if the dialogue act has a duration shorter than 1 second, 'no' otherwise. The value of 1 second has not been tested on our training corpus and is simply copied from the work of [15]. It is unclear how this feature might help in our setting, but it easily implemented and added for completeness.

5.1.3 Number of words in the current dialogue act

This feature counts the number of words in the current dialogue act. Just as in [15] the feature is a *nominal* one with three possible values: **one** (when there is 1 word in the dialogue act), **few** (when there are 2-4 words in the dialogue act), or **many** (when there are 5 or more words in the dialogue act).

5.1.4 Contains 1st person singular personal pronoun

This feature indicates whether or not one or more of the following first person singular Personal Pronouns occur within the dialogue act: *I*, *my*, *me*,

mine or *myself*.

5.1.5 Contains 1st person plural personal pronoun

This feature indicates whether or not one or more of the following first person plural Personal Pronouns occur within the dialogue act: *we*, *us*, *our*, *ours*, *ourselves* or *ourself*.

5.1.6 Contains 2nd person singular/plural personal pronoun

This feature indicates whether or not one or more of the following second person singular or plural Personal Pronouns occur within the dialogue act: *you*, *your*, *yours*, *yourselves* or *yourself*.

5.1.7 Contains 3rd person singular/plural personal pronoun

This feature indicates whether or not one or more of the following third person singular or plural Personal Pronouns occur within the dialogue act: *they*, *them*, *their*, *theirs*, *he*, *she*, *it*, *him*, *her*, *himself*, *herself*, *itself*, *themselves*, *hers* or *its*.

5.2 Context features

The following features are related to the context of the conversation. For those that look at a history window (5.2.3, 5.2.5 and 5.2.7), the optimal size of that window was determined by calculating the InfoGain of the feature with varying window sizes. To do this, the Information Gain Attribute Evaluator from WEKA was used. This method calculates the probability of an instance being addressed to the current speaker (prior probability) and compares this to the probability of being addressed given that a feature has a certain value. The higher the change in probability, the more information is gained from using that feature.

5.2.1 Leader role

The 'Leader Role' feature indicates whether you have the role of Project Manager, or discussion leader. The idea here is that this information may be useful because the project leader is the most active participant overall, so he may be addressed more often as well.

5.2.2 Type of previous dialogue act

The type of the dialogue act is one of the following classes: *Elicit*, *Social*, *Assess*, *Inform*, *Offer*, *Suggest* or *Comment-About-Understanding*. The dia-

logue act types from the corpus are mapped onto these seven types according to the mapping in Table 5.1.

5.2.3 Addressed history

The 'Addressed History' feature represents how often you have been addressed in the recent history of the conversation. The value is the number of dialogue acts that where addressed to you in a window of η dialogue acts. The number η has been determined to be optimal for $\eta = 6$ by varying η and calculating the InfoGain for every value (see Figure 5.1).

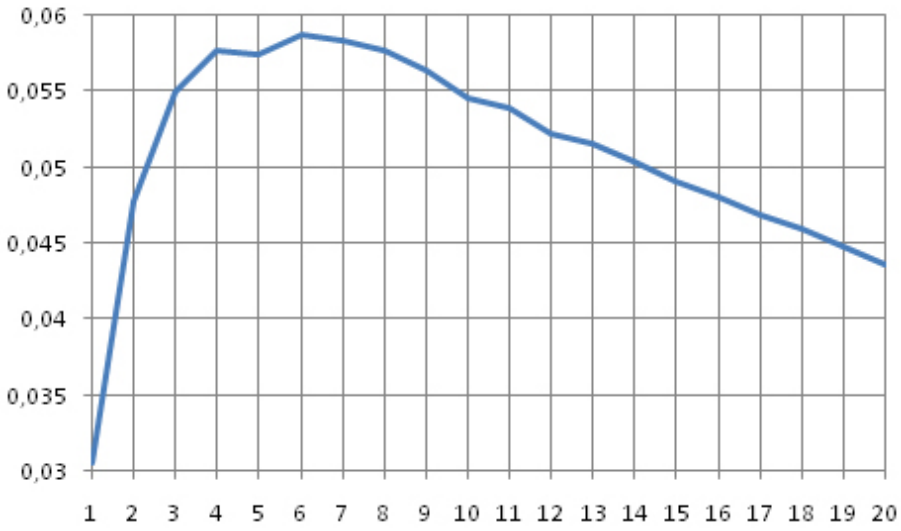


Figure 5.1: Window size η vs InfoGain of the Addressed History feature.

5.2.4 Previous dialogue act addressed to me

A simple 'yes' or 'no' feature that indicates whether or not the previous dialogue act was addressed to yourself. This is the same as the "Addressed History" feature above with a window of $\eta = 1$, except that the outcome would then be either 0 or 1, instead of 'yes' or 'no'.

5.2.5 Activity history

The 'Activity history' feature is a measure of your recent activity. The value is the number of dialogue acts that you have uttered yourself in a backward looking window of size η. The InfoGain for this feature is highest where $\eta = 5$ (see Figure 5.2).

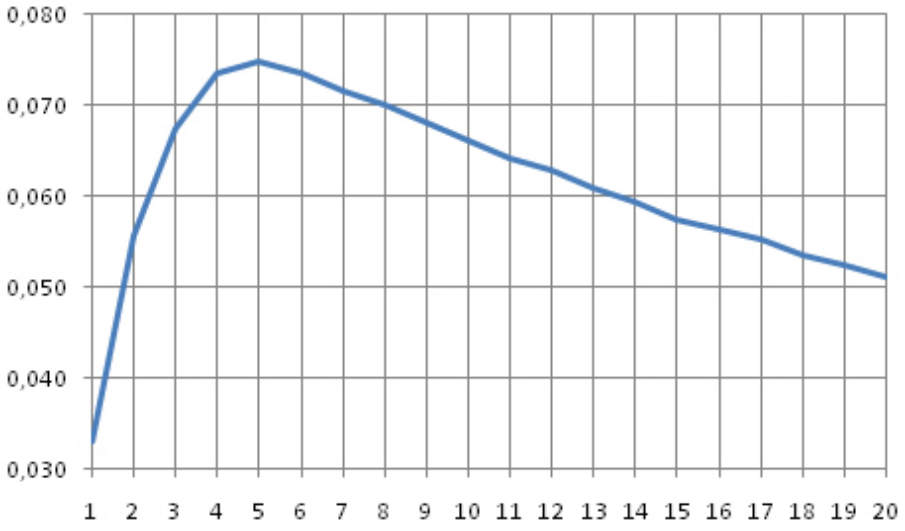


Figure 5.2: Window size η vs InfoGain of the Activity History feature.

5.2.6 Previous dialogue act uttered by me

A simple yes or no feature that indicates whether or not the previous dialogue act was uttered by yourself. This feature has the same value as the above "Activity history" feature with a window of $\eta = 1$.

5.2.7 Speaker diversity history

The 'Speaker diversity history' feature describes how many different speakers, other than yourself, have been active in the conversation over the course of the previous η dialogue acts. InfoGain for this feature peaks at $\eta = 3$ (see Figure 5.3). In the AMI corpus there are always 4 speakers, so the value for this feature is always either 0, 1 or 2. Should you want to use the system in a setting with more than 4 participants, the classifier will have to be retrained, because it can not handle values higher than 4. Therefore this feature may not translate very well to settings with more than 4 speakers.

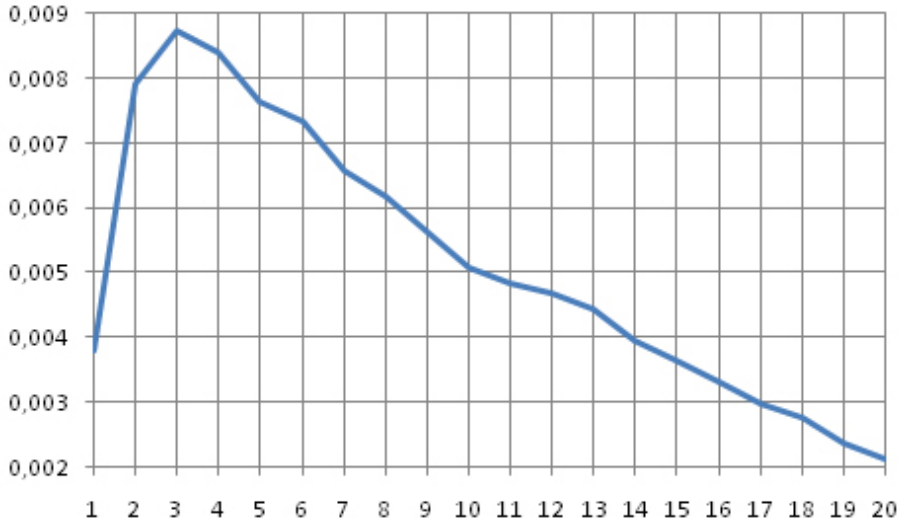


Figure 5.3: Window size η vs InfoGain of the Speaker Diversity History feature.

5.3 Results

Because the Linguistic- and Context Based Classifier uses addressee information as feature values, there are two possible ways to evaluate this classifier. The first one will be using the gold standard (hand annotated) values for the Addressing History features (5.2.3 and 5.2.4). The feature "Addresed History" for example, counts the number of times that you have been addressed in the previous η Dialogue Acts. This information is unknown however, because that is exactly what we are trying to predict. In the gold-standard case, we calculate the values of this feature based on the information in the corpus, disregarding the output of the classifier, so these two features will still always have the correct values. This evaluation gives us an idea of the hypothetical worth of these features. Because in a real time application, this information is not available, there is a second evaluation method. In the second, online feature evaluation, the output of the addressing classifier is used to compute the values for the next dialogue act features. So, if the classifier never classifies a Dialogue Act as being addressed to me, feature 5.2.4 will always be 'no' and feature 5.2.3 will always be 0. This is a more realistic evaluation for a real online demo setting. However, these evaluations still rely on perfect word- and dialogue act annotations (i.e. from the Gold Standard corpus).

We would like to find out what the optimal set of features and the best basic classification method for the task of predicting whether I am being

addressed is. Because there are 14 different features, a total of $2^{14} = 16384$ possible subsets of features will be tested for every classifier. We use a selection of rule-based-, tree-based-, and function based classifiers from the set used in [24] that are readily available in the WEKA toolkit, all with default classifier parameters. Where needed, the number of iterations or epochs is set to 20.

The baseline score for this classification task is relatively high. 89.2% of all dialogue acts in the test set are '*not addressed to me*' (remember: group addressed counts as *no*), so a classifier that labels every dialogue acts as such will receive that score. Therefore, all results that do not exceed this base score of 89.2% are ignored.

Figure 5.4 shows the results for the offline and online versions of the Linguistic- and Context Based Classifier. All results are reported for the specific best performing feature subset for that classifier, which may, and often does, differ for the offline and online versions. It is interesting to see that the order for the offline- and online results does not change.

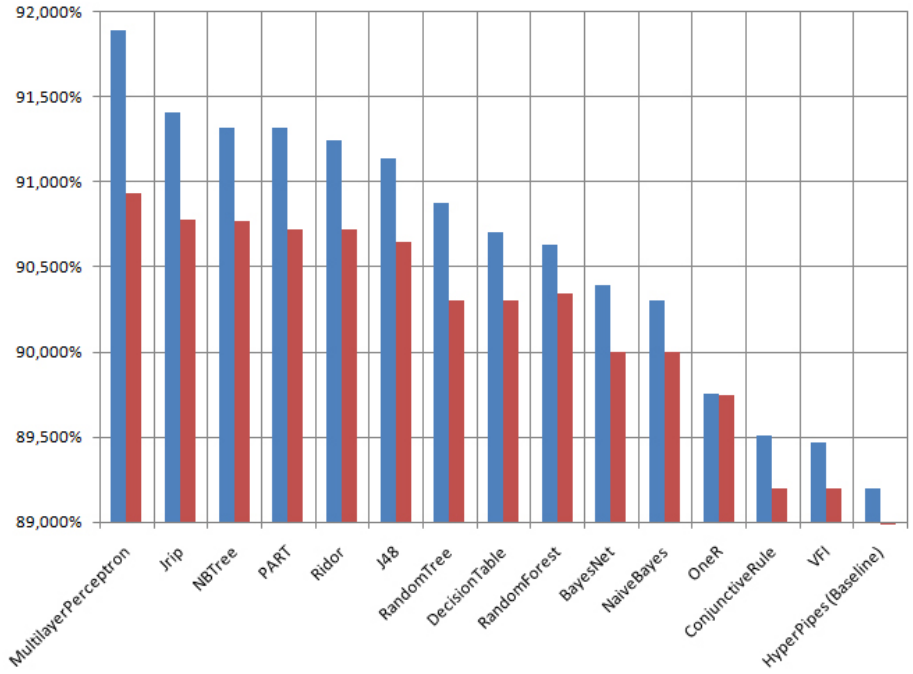


Figure 5.4: Results per Classifier for the Offline- (blue) and Online (red) versions of the Linguistic- and Context Based Classifier.

5.3.1 Optimal feature subsets

For the offline classifier, the best results are achieved with the Multilayer Perceptron classifier, with an accuracy score of 91.89%. The corresponding best-performing feature subset is given in Table 5.2.

Table 5.2: Optimal Feature set using the MultilayerPerceptron Classifier (Offline Scenario).

Paragraph	Feature
5.1.1	Type of the current Dialogue Act
5.1.3	Number of Words in the current Dialogue Act
5.1.4	Contains 1st person singular Personal Pronoun
5.1.5	Contains 1st person plural Personal Pronoun
5.1.6	Contains 2nd person singular/plural Personal Pronoun
5.1.7	Contains 3rd person singular/plural Personal Pronoun
5.2.1	Leader Role
5.2.2	Type of Previous Dialogue Act
5.2.3	Addressed History ($\eta = 6$)
5.2.4	Previous Dialogue Act addressed to me
5.2.5	Activity History ($\eta = 5$)
5.2.6	Previous Dialogue Act uttered by me
5.2.7	Speaker Diversity History ($\eta = 3$)

The best results for the online feature evaluation are also achieved with the Multilayer Perceptron classifier, with a score of 90.93% accuracy. The best-performing feature subset is given in Table 5.3.

Table 5.3: Optimal Feature set using the MultilayerPerceptron Classifier (Online Scenario).

Paragraph	Feature
5.1.1	Type of the current Dialogue Act
5.1.2	Short Dialogue Act
5.1.3	Number of Words in the current Dialogue Act
5.1.4	Contains 1st person singular Personal Pronoun
5.1.6	Contains 2nd person singular/plural Personal Pronoun
5.2.2	Type of Previous Dialogue Act
5.2.4	Previous Dialogue Act addressed to me
5.2.5	Activity History ($\eta = 5$)
5.2.6	Previous Dialogue Act uttered by me
5.2.7	Speaker Diversity History ($\eta = 3$)

5.3.2 Justification of parameters

In the list of fifteen classifiers tested above, the Multilayer Perceptron has a parameter to set the number of training epochs, and the RandomForest classifier has a parameter to set the number of trees. These two parameters have both been set to an arbitrary value of 20. We could not evaluate to an optimal value for every feature set because that would explode the search space of the problem. Therefore, we now take the best-performing feature subset from the experiments with default parameters above and vary these epocs and trees variables from 1 to 50 to see how justified our choice of 20 really was.

The results can be seen in Figure 5.5 for the MultiLayerPerceptron classifier and in Figure 5.6 for the RandomForest classifier.

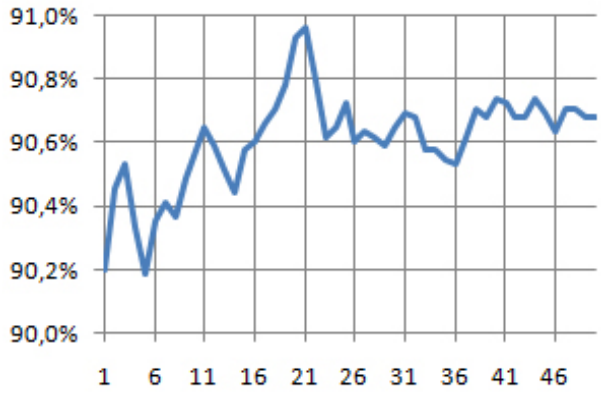


Figure 5.5: Results for the Online MultiLayerPerceptron classifier with best performing feature subset and varying number of traing epochs (horizontal axis).

For the MultiLayerPerceptron, the optimal result was gained with 21 training epochs with a score 90.96% accuracy versus 90.93% for 20 epochs (0.03% increase). For the RandomForest classifier, the optimal result was achieved with 16 trees, with a score of 90.39% versus 90.35% when using 20 trees (0.04% increase).

Although these findings do not definitely rule out the possibility that there is a parameter/feature-set combination that scores significantly better than any of the above results, it does show that our initial value of 20 seems reasonable and probably didn't lead to a severe underestimation of the classifier results.

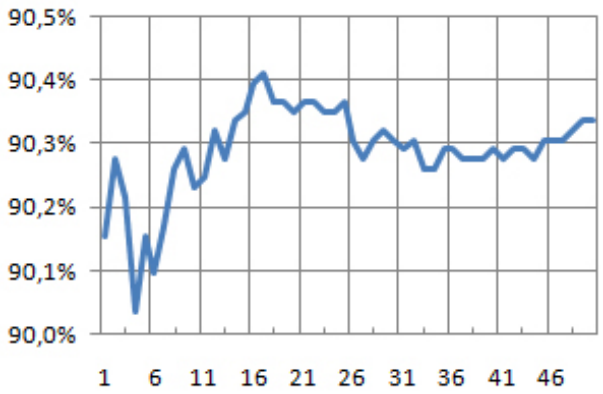


Figure 5.6: Results for the Online RandomForest classifier with best performing feature subset and varying number of trees in the forest (horizontal axis).

5.3.3 Discussion

Table 5.4 gives a summary of the results of the offline- and online experiments. The last row indicates how good the results are on the scale of the Base Score (89,24%) and the Hypothesized Maximum Score (92,47%) from Section 4.1.

Table 5.4: Summary of Results for the Linguistic- and Context Based Classifier

	Offline	Online
Accuracy:	91,89	90,93
Recall:	43,35	33,10
Precision:	70,18	66,02
F-Measure:	53,59	44,09
Achieved:	82%	53%

When comparing the accuracies of both classifiers, the difference does not seem to be that big. But when looking at the percentage of maximum achievable score, the difference becomes apparent. Unfortunately the results of the offline experiments can never be achieved in a real setting, because it uses hand-annotated addressing information to predict addressing, which is obviously cheating. What it does show is that it is useful to keep track of how often you have been addressed. Then, whenever the addressing software becomes better, these "addressing features" become more accurate, and this will boost the performance in return.

Tables 5.5 and 5.6 show the confusion matrices for both classification results. When we take 'yes' as a "positive" and 'no' as a "negative", the confusion tables should be read from left-to-right, top-to-bottom: true positives, false negatives, false positives, true negatives.

	Yes	No	Total
Yes	**313**	409	722
No	133	**5829**	5962
Total	446	6238	6684

Table 5.5: Confusion matrix for offline evaluation of the Linguistic- and Context Based Classifer.

	Yes	No	Total
Yes	**239**	483	722
No	123	**5839**	5962
Total	362	6322	6684

Table 5.6: Confusion matrix for online evaluation of the Linguistic- and Context Based Classifer.

In these confusion matrices you can see that only 313 and 239 times respectively out of a total of 6684 utterances the outcomes is 'yes', when it should be yes; or it **is** classified to *me*. Looking at the online version, 483 dialogue acts that were addressed to me are not classified as such, meaning, in a real application, I will not be notified that my attention is needed. This is still unacceptable performance for use in an application such as the User Engagement and Floor Control Demonstrator.

Looking further at the online version of the classifier, we see that many of the features (10 out of 14) are selected by the best-performing classifier, proving the the information of whether or not you are being addressed comes from many different ways of looking at the data. The results may not be satisfactory yet, but we still have the Visual Focus of Attention information as an untapped source of information. The next chapter will, in a similar way as this chapter, describe the creation of a classifier based on these type of features.

Chapter 6

Visual focus of attention classifier

Chapter 5 described an addressee classification method using only textual data. This section describes a second classifier that uses only visual data in the form of hand-annotated focus of attention information. In the next chapter, these two classifiers will be combined.

The focus of the speaker of an utterance can be a very important indicator of who he is addressing his speech to. But the focus of other participants can also be of use. We can distinguish three general issues with deriving classification features based on visual focus of attention: **who**, **when** and **how**?

- **Who**'s information do we use? Do we look at the speaker of the utterance alone, or at all participants in the meeting? Or just the non-active side participants?
- **When** exactly do we measure? Using the start and end times of a dialogue act as a window in which we look at the participant's focus of attention might not be the best method. A better window might include some time right before the dialogue act.
- **How** do we use our data exactly? One option is to feed exact timing information on how long a participant is looking at a possible addressee to the classifier. Another option is to calculate a threshold value, such that short glances at a participant are ignored. Only when the time that a person is looking at the possible addressee exceeds the threshold, we could count it as a visual focus on that participant.

For all features described below, the optimal window in which to look for focus of attention information may be different. Therefore, for every feature, the following experiments are done:

- First, simply the dialogue act duration is used as window to calculate the values of the feature. Then, a MultiLayerPerceptron is trained on that feature only, and evaluated on the test set. This result is the feature's default score.
- Secondly, the window in which to look is varied around the start time of the dialogue act (see Figure 6.1). Starting at $0, 0$ (empty window) and increasing the left and right windows individually by 1 second at a time in 20 steps. The right window can never exceed the end of the dialogue act[1]. For each of these 400 windows, the feature values are re-calculated, and a MultiLayerPerceptron classifier is trained and tested for that feature.
- The results of these experiments can be seen in the matrices for every feature (for example, see Table 6.1). In these tables, the positive values have a green background color, indicating improvements over the default score, and the negative values have red backgrounds, indicating a decline in performance. On the horizontal axis, the **right** window is shown, the vertical axis is the **left** window (e.g. down in the table is looking farther in the 'past').

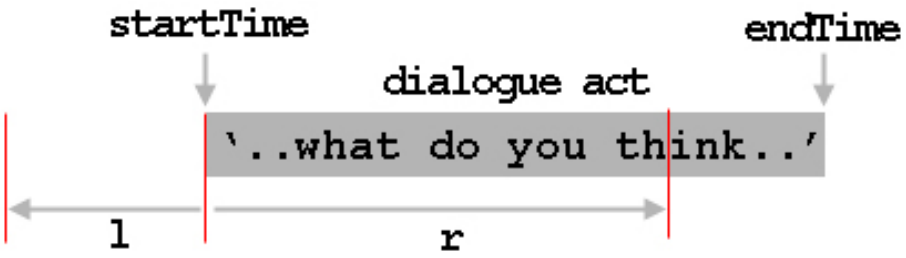


Figure 6.1: Dialogue act example with surrounding Left (l) and Right (r) windows.

6.1 Features

The following sections describe the features that are used for the Visual Focus of Attention Classifier.

[1] This is done as to never cheat and look into the future.

6.1.1 Total time everyone looks at me

The feature named "total time everyone looks at me" counts the total combined time that all other participants look at the participant currently under observation (me), during a given stretch of time around the dialogue act. This duration is normalized to the number of participants in the conversation other than the current participant. This means this feature has values between 0 (no other participant is looking at me) and $l+r$ (during the whole observed window, everyone looks at me, all the time).

The time frame analysis for this feature can be seen in Table 6.1. When using the start- and end times of the dialogue act as window for this feature, the result is 89.24% correctly classified instances. This is the absolute base score for the Visual Focus of Attention Classifier (e.g. all instances classified as 'no'). The values in Table 6.1 are relative to this score.

The highest result is at $(5, 1)$, by looking 5 seconds before the start of the dialogue act, to 1 second after the start of the dialogue act. The achieved result is $89.24\% + 0.63\% = 89.87\%$ classified correctly.

Table 6.1: *Window analysis for feature: 'total time everyone looks at me'. Scores are relative to the default score of 89.24%. The horizontal axis represents the right window whereas the vertical axis represents the left.*

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	0,00	0,02	0,26	0,11	0,19	-0,26	-0,09	-0,22	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1	-0,11	0,26	0,41	0,17	0,17	0,00	-0,17	-1,19	-1,06	-1,26	-0,87	0,00	0,00	-0,35	0,00	-0,24	-0,22	-0,22	-0,22	-0,45
2	0,37	0,39	0,19	0,09	0,06	-0,09	-1,08	-0,84	-0,84	-0,89	-0,84	-0,82	-1,02	-0,32	-0,32	0,00	-0,32	-0,28	-0,28	-0,28
3	0,43	0,41	0,45	0,37	0,19	0,19	-0,09	-1,54	-1,43	-1,45	-1,41	-1,36	-1,43	-1,30	-1,32	-1,30	-1,32	-1,04	-1,04	-1,04
4	0,56	0,58	0,54	0,43	0,58	0,48	0,41	0,37	-1,93	-1,82	-1,88	-1,97	-1,99	-1,86	-1,88	-1,90	-1,90	-1,90	-1,90	-1,90
5	0,48	0,63	0,61	0,56	0,32	0,22	0,26	0,13	-0,02	-2,21	-2,21	-2,32	-2,38	-2,40	-2,42	-2,29	-2,42	-2,42	-2,42	-2,42
6	0,45	0,48	0,43	0,61	-0,04	-0,04	-0,11	-0,06	-0,11	-2,84	-2,81	-2,86	-2,92	-2,94	-2,86	-2,86	-2,90	-2,90	-2,90	-2,97
7	0,41	0,41	0,56	0,19	0,26	-0,30	-0,39	-0,52	-0,48	-0,50	-0,52	-3,35	-3,42	-3,46	-3,46	-3,29	-3,29	-3,53	-3,53	-3,53
8	0,30	0,37	0,39	0,43	0,30	0,19	0,17	-0,93	-0,91	-0,84	-0,84	-0,84	-3,74	-3,74	-3,79	-3,59	-3,61	-3,79	-3,79	-3,79
9	0,28	0,30	0,35	0,43	0,39	0,30	0,28	0,26	-1,08	-1,10	-1,08	-1,08	-1,17	-1,17	-1,08	-1,08	-1,08	-1,10	-1,10	-1,10
10	0,26	0,19	0,24	0,37	0,17	0,15	0,13	0,04	-1,10	-1,17	-1,15	-1,15	-1,19	-1,19	-1,19	-1,15	-1,15	-1,15	-1,15	-1,17
11	0,19	0,13	0,30	0,35	0,43	0,24	0,24	0,24	0,13	0,04	0,04	-1,06	-1,06	-1,06	-1,06	-1,06	-1,06	-1,06	-0,97	-0,97
12	0,00	0,30	0,35	0,41	0,43	0,06	0,09	0,06	0,04	0,04	-1,19	0,32	-1,26	-1,23	-1,23	-1,23	-1,23	-1,23	-1,23	-1,23
13	0,00	0,09	0,35	0,43	0,41	0,43	-0,06	-0,17	-0,09	-0,09	-0,11	-0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
14	0,19	0,15	-0,04	0,30	0,26	0,28	-0,09	-0,22	-0,26	-0,09	-0,15	-0,97	-1,00	-0,13	-1,04	-1,04	-1,04	-1,04	-0,13	-0,13
15	0,00	0,19	0,22	0,26	0,19	-0,13	-0,17	-0,32	0,09	-0,19	-0,19	-0,22	-0,19	-0,93	-0,93	-0,93	-0,93	-0,93	-0,93	-0,93
16	0,17	0,04	0,11	0,11	0,11	0,13	0,15	-0,30	-0,30	-0,15	-0,37	-0,22	-0,22	-0,19	-0,19	-0,19	-0,19	-0,19	-0,19	-0,19
17	0,00	0,04	0,17	0,09	0,11	0,11	0,11	0,11	0,06	0,06	0,02	-0,17	-0,19	-0,19	-0,17	-0,17	-0,17	-0,17	-0,17	-0,17
18	0,06	0,11	0,13	0,00	0,02	-0,04	0,04	-0,06	-0,17	0,00	0,00	-0,02	-0,06	-0,06	-0,06	-0,04	-0,04	-0,04	-0,04	-0,04
19	0,00	0,00	0,00	0,15	0,00	0,00	0,00	0,00	-0,11	-0,04	0,04	-0,04	-0,02	-0,06	-0,04	-0,04	-0,04	-0,04	-0,04	-0,04

6.1.2 Total time everyone looks at me (normalized)

The feature named "total time everyone looks at me (normalized)" is calculated in the same way as the feature above (Section 6.1.1), only this time its value is normalized to the duration of the window. So all values are divided by $l + r$, and the values all lie between 0 (no other participant is looking at me) and 1 (during the whole observed window, everyone looks at me, all the time).

The time frame analysis for this feature can be seen in Table 6.2. When using the start- and end times of the dialogue act as window for this feature, the result is 89.91% correctly classified instances. This is already better than the non-normalized version of this feature (6.1.1). The values in Table 6.2 are relative to this score.

The highest results are at $(3, 7)$ and increasing right windows, by looking 3 seconds before the start of the dialogue act, to 7 or more seconds after the start of the dialogue act. The achieved result is $89.91\% + 0.35\% = 90.26\%$ classified correctly.

Table 6.2: *Window analysis for feature: 'total time everyone looks at me (normalized)'. Scores are relative to the default score of 89.91%. The horizontal axis represents the right window whereas the vertical axis represents the left.*

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	-0,67	-0,24	0,06	0,06	0,04	0,04	0,02	0,04	0,04	0,00	0,04	0,00	0,04	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1	-0,78	-0,50	0,19	0,17	0,24	0,17	0,17	0,22	0,22	0,24	0,22	0,22	0,24	0,24	0,24	0,24	0,24	0,24	0,24	0,24
2	-0,30	-0,11	0,09	0,13	0,22	0,26	0,24	0,26	0,24	0,26	0,24	0,24	0,24	0,24	0,24	0,24	0,24	0,24	0,24	0,24
3	-0,26	-0,15	0,17	0,24	0,30	0,32	0,32	0,35	0,35	0,35	0,35	0,35	0,35	0,35	0,35	0,35	0,35	0,35	0,35	0,35
4	-0,11	-0,28	0,11	0,15	0,19	0,26	0,24	0,24	0,24	0,24	0,24	0,24	0,24	0,24	0,24	0,24	0,24	0,24	0,24	0,24
5	-0,19	-0,02	0,22	0,06	0,11	0,17	0,19	0,22	0,17	0,17	0,17	0,17	0,17	0,17	0,17	0,17	0,17	0,17	0,17	0,17
6	-0,22	-0,26	0,02	0,00	0,00	-0,02	-0,04	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11
7	-0,26	-0,32	-0,06	-0,35	-0,06	-0,06	-0,09	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13
8	-0,37	-0,32	0,02	0,04	0,02	-0,02	0,00	-0,04	-0,04	-0,04	-0,04	-0,04	-0,04	-0,04	-0,04	-0,04	-0,04	-0,04	-0,04	-0,04
9	-0,39	-0,43	-0,35	-0,22	-0,17	-0,17	-0,24	-0,24	-0,24	-0,24	-0,24	-0,24	-0,24	-0,24	-0,24	-0,24	-0,24	-0,24	-0,24	-0,24
10	-0,41	-0,37	-0,11	-0,17	-0,15	-0,15	-0,17	-0,17	-0,17	-0,17	-0,17	-0,17	-0,17	-0,17	-0,17	-0,17	-0,17	-0,17	-0,17	-0,17
11	-0,48	-0,41	-0,43	-0,43	-0,43	-0,43	-0,45	-0,45	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43
12	-0,67	-0,39	-0,41	-0,43	-0,43	-0,45	-0,45	-0,45	-0,45	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43
13	-0,67	-0,58	-0,50	-0,39	-0,39	-0,41	-0,43	-0,41	-0,41	-0,41	-0,41	-0,41	-0,41	-0,41	-0,41	-0,41	-0,41	-0,41	-0,41	-0,41
14	-0,48	-0,50	-0,45	-0,45	-0,45	-0,45	-0,45	-0,45	-0,45	-0,45	-0,45	-0,45	-0,45	-0,45	-0,45	-0,45	-0,45	-0,45	-0,45	-0,45
15	-0,67	-0,48	-0,52	-0,48	-0,48	-0,54	-0,54	-0,48	-0,48	-0,48	-0,48	-0,48	-0,48	-0,48	-0,48	-0,48	-0,48	-0,48	-0,48	-0,48
16	-0,50	-0,67	-0,45	-0,43	-0,45	-0,43	-0,67	-0,67	-0,67	-0,67	-0,67	-0,67	-0,67	-0,67	-0,67	-0,67	-0,67	-0,67	-0,67	-0,67
17	-0,67	-0,48	-0,45	-0,45	-0,45	-0,45	-0,45	-0,54	-0,54	-0,54	-0,54	-0,54	-0,54	-0,54	-0,54	-0,54	-0,54	-0,54	-0,54	-0,54
18	-0,61	-0,61	-0,45	-0,43	-0,50	-0,50	-0,50	-0,50	-0,50	-0,50	-0,50	-0,50	-0,50	-0,50	-0,50	-0,50	-0,50	-0,50	-0,50	-0,50
19	-0,67	-0,67	-0,67	-0,63	-0,65	-0,63	-0,61	-0,61	-0,61	-0,61	-0,61	-0,61	-0,61	-0,61	-0,61	-0,61	-0,61	-0,61	-0,61	-0,61

6.1.3 Total time speaker looks at me

The feature named "total time speaker looks at me" counts the total time that the speaker of the current dialogue act looks at the participant currently under observation (me), during a given stretch of time around the dialogue act. This means this feature has values between 0 (speaker is not looking at me at all) and $l + r$ (during the whole observed window, the speaker looks at me all the time).

The time frame analysis for this feature can be seen in Table 6.3. When using the start- and end times of the dialogue act as window for this feature, the result is 89.24% correctly classified instances. This is the absolute base score for the Visual Focus of Attention Classifier (e.g. all instances classified as 'no'). The values in Table 6.3 are relative to this score.

The highest result is at $(3, 1)$, by looking 3 seconds before the start of the dialogue act, to 1 second after the start of the dialogue act. The achieved result is $89.24\% + 0.93\% = 90.17\%$ classified correctly.

Table 6.3: *Window analysis for feature: 'total time speaker looks at me'. Scores are relative to the default score of 89.24%. The horizontal axis represents the right window whereas the vertical axis represents the left.*

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	0,00	0,11	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1	0,00	0,41	0,04	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
2	0,35	0,41	0,56	0,30	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
3	0,65	0,93	0,71	0,37	0,24	0,15	0,04	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
4	0,61	0,52	0,37	0,58	0,17	0,26	-0,17	-0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,13	0,00	0,00	0,00	0,00
5	0,58	0,63	0,67	0,54	0,13	0,06	-0,22	0,04	0,09	0,00	-0,56	0,00	0,00	0,00	0,00	0,00	0,00	0,37	0,37	0,37
6	0,54	0,48	0,54	0,41	0,56	0,11	0,39	0,09	-0,35	0,48	0,11	0,56	0,13	0,00	0,00	0,00	0,00	-0,04	-0,04	-0,04
7	0,52	0,61	0,67	0,63	0,54	0,56	0,43	0,45	0,45	0,04	0,17	0,19	0,19	0,00	0,43	0,54	0,11	-0,45	-0,45	-0,45
8	0,45	0,56	0,48	0,48	0,54	0,50	0,30	0,26	0,17	0,30	0,28	0,24	0,32	0,24	0,24	0,24	0,30	0,30	0,30	0,30
9	0,50	0,69	0,48	0,43	0,39	0,52	0,48	0,39	0,24	0,26	0,28	0,35	0,39	0,19	0,35	0,15	0,15	0,13	0,13	0,13
10	0,43	0,52	0,54	0,41	0,48	0,37	0,30	0,43	0,39	0,28	0,32	0,35	0,32	0,39	0,37	0,11	0,11	0,11	0,11	0,11
11	0,45	0,45	0,48	0,56	0,30	0,63	0,45	0,50	0,11	0,11	0,19	0,26	0,24	0,24	0,09	0,00	0,09	0,09	0,09	0,09
12	0,45	0,50	0,61	0,52	0,52	0,11	0,56	0,48	0,11	0,19	0,22	0,19	0,22	0,00	0,22	0,11	0,00	0,00	0,00	0,00
13	0,48	0,52	0,54	0,50	0,52	0,52	0,50	0,41	0,43	0,45	0,28	0,26	0,28	0,22	-0,37	-0,37	0,28	-0,30	-0,30	-0,30
14	0,61	0,56	0,52	0,56	0,43	0,50	0,48	0,28	0,24	0,41	0,26	0,26	0,24	0,17	-0,43	0,19	0,19	-0,35	-0,35	-0,35
15	0,48	0,50	0,65	0,48	0,50	0,45	0,39	0,41	0,22	0,24	0,26	0,26	0,22	0,28	0,13	0,13	-0,61	-0,56	-0,56	-0,56
16	0,54	0,67	0,58	0,52	0,58	0,43	0,41	0,43	0,19	0,22	0,22	0,22	0,22	0,22	0,06	0,19	-0,61	-0,61	-0,61	-0,61
17	0,50	0,67	0,69	0,67	0,48	0,58	0,54	0,58	0,19	0,19	0,15	0,19	0,17	0,17	0,19	0,22	-0,67	-0,67	-0,67	-0,67
18	0,63	0,65	0,82	0,74	0,50	0,69	0,65	0,63	0,13	0,15	-0,06	0,02	0,15	0,15	0,15	-0,15	0,04	-0,50	-0,50	-0,50
19	0,54	0,69	0,78	0,39	0,63	0,71	0,71	0,63	0,58	0,06	0,06	0,11	0,11	0,11	0,11	0,13	0,04	0,13	0,13	0,13

6.1.4 Total time speaker looks at me (normalized)

The feature named "total time speaker looks at me (normalized)" is calculated in the same way as the feature above (Section 6.1.3), only this time its value is normalized to the duration of the window. So all values are divided by $l + r$, and the values all lie between 0 (speaker is not looking at me at all) and 1 (during the whole observed window, the speaker looks at me all the time).

The time frame analysis for this feature can be seen in Table 6.4. When using the start- and end times of the dialogue act as window for this feature, the result is 90.28% correctly classified instances. This is already better than the non-normalized version of this feature (6.1.3). The values in Table 6.4 are relative to this score.

The highest result is again at $(3, 1)$, by looking 3 seconds before the start of the dialogue act, to 1 second after the start of the dialogue act. The achieved result is $90.28\% + 0.15\% = 90.43\%$ classified correctly.

Table 6.4: *Window analysis for feature: 'total time speaker looks at me (normalized)'. Scores are relative to the default score of 90.28%. The horizontal axis represents the right window whereas the vertical axis represents the left.*

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	-1,04	-0,43	0,00	0,06	0,04	0,04	0,02	0,00	0,02	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1	-1,04	-0,24	-0,43	-0,24	-0,37	-0,35	-0,41	-0,45	-0,45	-0,28	-0,30	-0,28	-0,28	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43
2	-0,69	-0,24	-0,11	-0,02	0,09	0,02	0,06	0,00	0,02	0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
3	-0,41	0,15	-0,24	-0,13	-0,06	-0,06	0,00	0,13	-0,06	-0,06	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11
4	-0,43	-0,17	-0,15	-0,19	-0,09	-0,09	-0,06	-0,13	-0,11	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13
5	-0,45	-0,19	-0,19	-0,17	-0,17	-0,22	-0,19	-0,19	-0,19	-0,17	-0,19	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22
6	-0,50	-0,50	-0,24	-0,02	-0,15	-0,13	-0,09	-0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02	0,02
7	-0,52	-0,35	-0,17	-0,17	-0,11	-0,09	-0,06	-0,06	-0,09	-0,04	-0,04	-0,04	-0,06	-0,06	-0,06	-0,06	-0,06	-0,06	-0,06	-0,06
8	-0,58	-0,43	-0,28	-0,22	-0,15	-0,13	-0,11	-0,09	-0,09	-0,06	-0,06	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13
9	-0,54	-0,26	-0,30	-0,17	-0,15	-0,17	-0,15	-0,15	-0,17	-0,17	-0,17	-0,17	-0,17	-0,17	-0,17	-0,17	-0,17	-0,17	-0,17	-0,17
10	-0,61	-0,52	-0,54	-0,37	-0,50	-0,45	-0,43	-0,50	-0,50	-0,50	-0,50	-0,50	-0,50	-0,50	-0,50	-0,50	-0,50	-0,50	-0,50	-0,50
11	-0,58	-0,56	-0,48	-0,41	-0,48	-0,39	-0,41	-0,48	-0,41	-0,37	-0,37	-0,39	-0,39	-0,39	-0,48	-0,48	-0,48	-0,45	-0,45	-0,48
12	-0,58	-0,52	-0,48	-0,48	-0,41	-0,45	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43	-0,43
13	-0,56	-0,52	-0,50	-0,37	-0,37	-0,35	-0,32	-0,30	-0,30	-0,30	-0,30	-0,30	-0,30	-0,30	-0,30	-0,30	-0,30	-0,30	-0,30	-0,30
14	-0,43	-0,39	-0,28	-0,41	-0,26	-0,26	-0,24	-0,24	-0,24	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22
15	-0,56	-0,52	-0,35	-0,45	-0,39	-0,43	-0,43	-0,35	-0,32	-0,32	-0,32	-0,32	-0,32	-0,32	-0,32	-0,32	-0,32	-0,32	-0,32	-0,32
16	-0,50	-0,32	-0,35	-0,32	-0,26	-0,24	-0,26	-0,24	-0,24	-0,24	-0,24	-0,24	-0,24	-0,24	-0,24	-0,24	-0,24	-0,24	-0,24	-0,24
17	-0,54	-0,37	-0,35	-0,32	-0,32	-0,32	-0,35	-0,35	-0,35	-0,35	-0,35	-0,35	-0,35	-0,35	-0,35	-0,35	-0,35	-0,35	-0,35	-0,35
18	-0,41	-0,37	-0,24	-0,17	-0,17	-0,17	-0,15	-0,15	-0,15	-0,15	-0,15	-0,15	-0,15	-0,15	-0,15	-0,15	-0,15	-0,15	-0,15	-0,15
19	-0,50	-0,41	-0,28	-0,28	-0,22	-0,22	-0,19	-0,19	-0,19	-0,19	-0,19	-0,19	-0,19	-0,19	-0,19	-0,19	-0,19	-0,19	-0,19	-0,19

6.1.5 Speaker looks at me (yes/no)

The feature "speaker looks at me (yes/no)" takes the values from the previous feature (Section 6.1.4) and converts this into a binary feature using a threshold. If the value, which is normalized and lies between 0 and 1, exceeds a certain threshold τ, the feature has the value **yes**, otherwise, the value is **no**. All values have been calculated with the optimal focus of attention window of $(3, 1)$ as calculated in 6.1.4.

Figure 6.2 shows the threshold value plotted against the classification result when using that threshold.

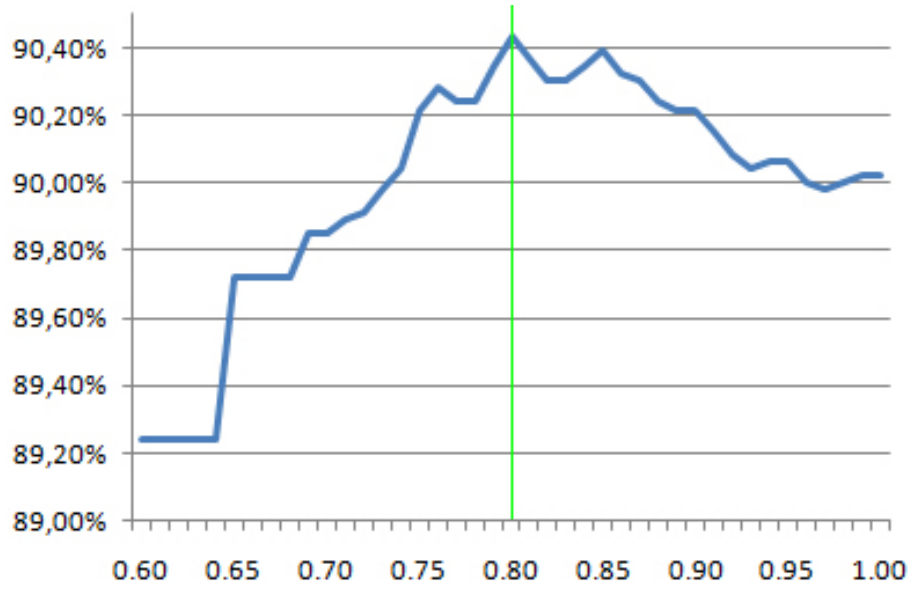


Figure 6.2: Threshold values of 0.6 and up (horizontal axis) versus classifier performance (vertical axis) for the 'speaker looks at me (yes/no)' feature. All values for thresholds lower than 0.6 are the same as 0.6.

The maximum result is achieved with a threshold of 0.80, with a score of 90.43%. This is the same maximum score as the previous feature 6.1.4.

6.1.6 Total time side participants look at me

The feature named "total time side participants look at me" counts the total combined time that the participants other than the one currently under observation, and the speaker of the dialogue act (e.g. the side participants) look at me, during a given stretch of time around the dialogue act.

This duration is normalized to the maximum number of possible side participants in the conversation (total number of participants minus 2 for the speaker and me). This means this feature has values between 0 (no side participant is looking at me) and $l + r$ (during the whole observed window, all side participants are looking at me, all the time).

The time frame analysis for this feature can be seen in Table 6.5. When using the start- and end times of the dialogue act as window for this feature, the result is 89.24% correctly classified instances. This is the absolute base score for the Visual Focus of Attention Classifier (e.g. all instances classified as 'no'). The values in Table 6.5 are relative to this score.

The highest result is at $(3, 0)$, by looking 3 seconds before the start of the dialogue act up to the start time of the dialogue act. The achieved result is $89.24\% + 0.19\% = 89.44\%$ classified correctly.

Table 6.5: *Window analysis for feature: 'total time side participants looks at me'. Scores are relative to the default score of 89.24%. The horizontal axis represents the right window whereas the vertical axis represents the left.*

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1	0,00	0,09	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
2	0,00	-0,02	-0,19	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
3	0,19	0,00	-0,02	-0,09	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
4	0,00	0,00	0,00	0,00	-0,39	0,00	0,00	0,00	0,00	0,00	0,00	0,00	-1,47	0,00	0,00	0,00	0,00	0,00	0,00	0,00
5	0,00	0,02	0,00	0,00	-0,32	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
6	0,00	0,15	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
7	0,00	0,13	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
8	0,00	0,00	-0,06	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
9	0,04	0,11	0,00	0,00	-0,13	0,00	0,00	-0,61	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
10	-0,02	0,00	0,00	0,00	0,00	0,00	0,00	-0,04	0,00	0,00	0,00	0,00	-0,11	0,00	0,00	0,00	0,00	0,00	0,00	0,00
11	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	-0,04	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
12	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
13	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
14	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	-0,50	-0,17	0,00	0,00	0,00	0,00	0,00	0,00	0,00
15	0,00	0,00	-0,09	-0,06	-0,11	-0,22	0,00	0,00	0,00	0,00	0,00	0,00	-0,41	-0,41	-0,41	-0,41	-0,41	-0,41	-0,41	-0,41
16	-0,04	0,00	-0,02	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,06	0,06	0,06	-0,04	-0,04	-0,04	-0,04	-0,04
17	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09	0,09
18	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,02	-0,28	-0,28	0,02	0,02	0,02	0,02	0,02	0,02	0,09	0,02	0,02
19	0,00	0,00	0,00	-0,09	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00

6.1.7 Total time side participants look at me (normalized)

The feature named "total time side participants look at me (normalized)" is calculated in the same way as the feature above (Section 6.1.6), only this time its value is normalized to the duration of the window. So all values are divided by $l + r$, and the values all lie between 0 (no side participants are looking at me at all) and 1 (during the whole observed window, all side participants look at me all the time).

The time frame analysis for this feature can be seen in Table 6.6. When using the start- and end times of the dialogue act as window for this feature, the result is 89.46% correctly classified instances. This is already a higher score than the non-normalized version of this feature (6.1.6). The values in Table 6.6 are relative to this score.

The only scores that are not lower than the default score are those at $(0, 7)$ and increasing right windows. Because these scores are exactly the same as the default score, the best achieved result remains: 89.46%.

Table 6.6: *Window analysis for feature: 'total time side participants looks at me (normalized)'. Scores are relative to the default score of 89.46%. The horizontal axis represents the right window whereas the vertical axis represents the left.*

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
0	-0,22	-0,22	-0,22	-0,15	-0,04	-0,02	-0,13	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
1	-0,22	-0,22	-0,22	-0,09	-0,13	-0,13	-0,13	-0,11	-0,15	-0,11	-0,11	-0,11	-0,15	-0,11	-0,15	-0,15	-0,15	-0,15	-0,15	-0,15
2	-0,22	-0,13	-0,28	-0,22	-0,28	-0,28	-0,28	-0,28	-0,28	-0,28	-0,28	-0,28	-0,28	-0,28	-0,28	-0,28	-0,28	-0,28	-0,28	-0,28
3	-0,02	-0,22	-0,02	-0,22	-0,04	-0,04	-0,11	-0,04	-0,11	-0,04	-0,04	-0,04	-0,04	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11
4	-0,22	-0,22	-0,13	-0,32	-0,32	-0,26	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11	-0,11
5	-0,22	-0,06	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22
6	-0,22	-0,22	-0,22	-0,17	-0,17	-0,04	-0,19	-0,17	-0,17	-0,22	-0,22	-0,22	-0,22	-0,17	-0,17	-0,22	-0,22	-0,22	-0,22	-0,17
7	-0,22	-0,22	-0,22	-0,35	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22
8	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22
9	-0,17	-0,09	-0,04	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22
10	-0,24	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22
11	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22
12	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22
13	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22
14	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22
15	-0,22	-0,22	-0,37	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22
16	-0,26	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22
17	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22
18	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22
19	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22	-0,22

6.1.8 Number of participants looking at me

The feature named "number of participants looking at me" represents the number of other participants in the meeting that are looking at me. To calculate this, there are two variables: the optimal window in which to look, and the threshold for determining when we say that a participant is looking at 'me'.

For every participant in the meeting, other than myself, the percentage of the window that he/she is looking at me is calculated. If this percentage exceeds the threshold τ, this participant is added to the count. This amounts in a feature value between 0 (no participants are looking at me) and $P_{max}-1$ (the number of participants in the meeting, minus myself).

Because we now need to vary three variables (left-window, right-window and threshold), the amount of experiments that need to be done might explode. Because in the window analysis of the previous features we have seen that left- and right window sizes of 10 and above do not generate good results for any of the features, we take 10 as the maximum value in the window analysis here. For the threshold values, we go from 0 to 1 in 10 steps, so we have a manageable, total number of $10 * 10 * 10 = 1000$ experiments to do.

Of these 1000 experiments, the 15 best results where all achieved with a threshold of 0.9. Therefore, we assume this to be the optimal threshold value. Table 6.7 shows the results of the window analysis with $\tau = 0.9$. The default score with $\tau = 0.9$ and the standard start- and end times of the dialogue act as window borders is 89.72%. The highest results are achieved at $(3, 2)$ and $(3, 3)$ with a total value of $89.72\% + 0.43\% = 90.15\%$ correctly classified.

Table 6.7: *Window analysis for feature: 'number of participants looking at me'. Scores are relative to the default score of 89.72%. All scores are calculated with a Threshold of* $\tau = 0.9$*. The horizontal axis represents the right window whereas the vertical axis represents the left.*

	0	1	2	3	4	5	6	7	8	9
0	-0,48	-0,02	0,02	0,04	0,00	-0,48	-0,13	-0,48	-0,48	-0,09
1	-0,39	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13	-0,13
2	-0,17	-0,15	-0,17	-0,17	-0,17	-0,28	-0,28	-0,28	-0,28	-0,28
3	-0,02	-0,11	0,43	0,43	0,39	0,26	0,28	0,26	0,26	0,26
4	-0,24	-0,37	0,37	0,24	0,26	0,26	0,26	0,24	0,26	0,26
5	-0,48	-0,39	0,30	0,13	0,13	0,15	0,02	-0,39	-0,39	0,04
6	-0,32	-0,48	-0,02	-0,28	-0,30	-0,48	-0,48	-0,48	-0,48	-0,48
7	-0,41	-0,48	-0,19	-0,22	-0,24	-0,43	-0,43	-0,41	-0,41	-0,41
8	-0,41	-0,41	-0,41	-0,24	-0,41	-0,41	-0,41	-0,41	-0,41	-0,41
9	-0,28	-0,26	-0,26	-0,26	-0,41	-0,41	-0,41	-0,41	-0,41	-0,41

Figure 6.3 shows the scores with a window of $(3, 3)$ and varying threshold values.

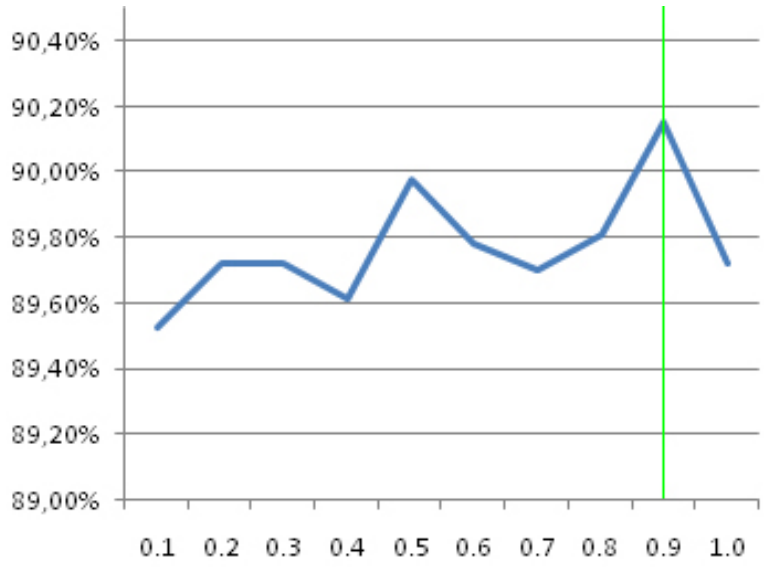


Figure 6.3: Threshold analysis for the 'number of participants looking at me' feature, using a window of $(3, 3)$. Threshold values of 0.1 and up (horizontal axis) versus classifier performance (vertical axis).

6.2 Results

In the previous subsections, each different feature has been optimized individually by testing classifiers with different feature parameter values. Now, we can set the parameters of every feature to their optimal values and test combinations of features. Again, we train and test a classifier with every possible subset of features. In this case, $2^8 = 256$ different subsets (including the empty set, which is useless of course).

During the feature optimization we used a MultiLayerPerceptron classifier, because this was the best candidate from the results in Section 5.3. It would have simply been way too time consuming to do the feature parameter optimization with a number of different classifiers. However, for the feature subset analysis, we do not assume that the MultiLayerPerceptron is the best option again. Therefore, we train and test a number of classifiers, to see how the results vary among them. Because the number of feature subsets that have to be tested is much less than with the Linguistic- and Context Based classifier, we now use the same 30 classifiers as used in [24] to train and test the FOA-based features.

The results of these experiments can be seen in Figure 6.4. As you can see, the MultiLayerPerceptron, or Neural Network, classifier scores very high again. The NBTree (decision tree with Naive Bayes classifiers at the leaves) and IbK (K-nearest neighbour) algorithms perform slightly better (90.84%, versus 90.80% for the MLP) than the MLP, but for simplicity's sake and because the increase is only so little (0.04%), we will continue using the Neural Network classifier in this work.

Although the Visual Focus of Attention classifier performed slighly worse than the Linguistic and Context Based classifier from the previous chapter, there still is a crucial step to be taken to improve the scores. The next chapter will deal with combining the two different classifiers in a number of different ways.

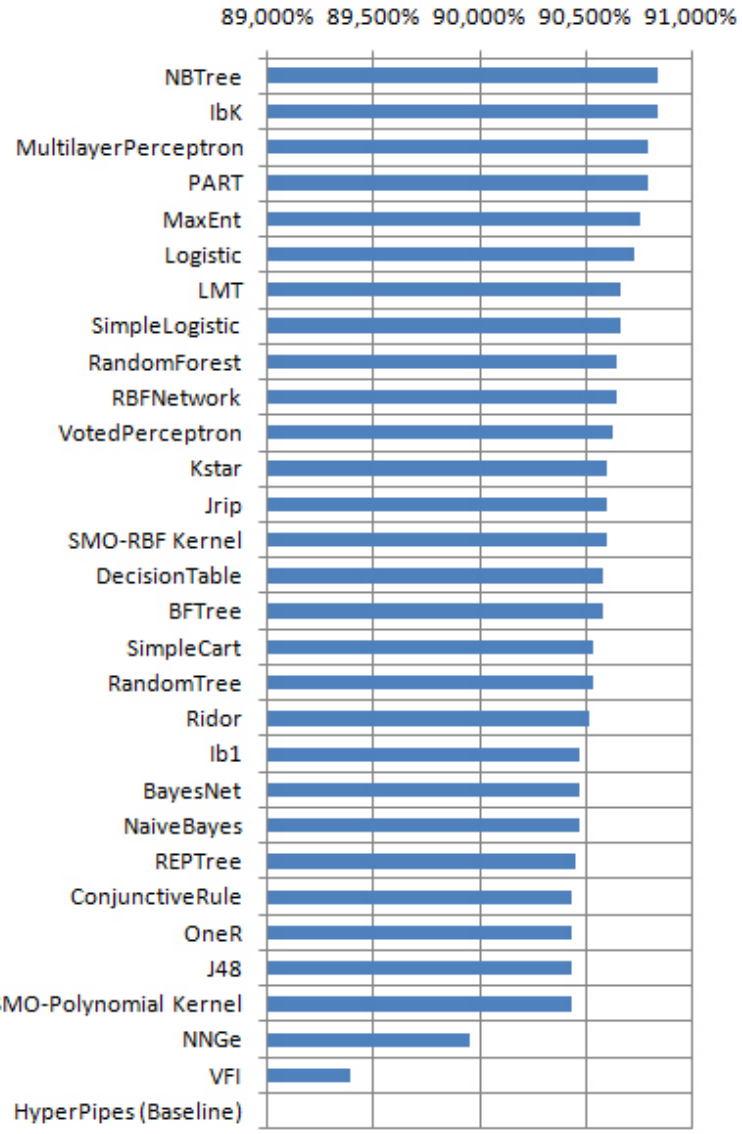


Figure 6.4: Classification results of 30 different classifiers using the optimal feature subset for each classifier.

Chapter 7

Results of the combined classifiers

In this section we look at how we can combine the results from the Linguistic- and Context Based Classifier and the Visual Focus of Attention Classifier. For easy reference, the results of both classifiers are repeated here first. For the Linguistic classifier, we take the scores for the online version, because that is the one that can be used in a real-time system (see Chapter 5).

Table 7.1: Summary of Results for the Linguistic- and Context Based Classifier.

# Train Instances	21873
# Test Instances	6684
Base score	89,20%
Achieved score	90,93%
Maximum score	92,47%
% Of maximum	53%

Table 7.2: Summary of Results for the Visual Focus of Attention Classifier.

# Train Instances	15030
# Test Instances	4620
Base score	89,24%
Achieved score	90,80%
Maximum score	92,47%
% Of maximum	48%

Now we want to know what we can achieve if we use both linguistic and visual information. There are a number of different ways to do this. The following three sections describe three different approaches. In the first one,

we simply use all the features described in 5 and 6 and train a classifier on all of them. The second approach is to take the output of both classifiers as training- and test data for a third classifier, which will determine the final decision based on both classifiers. The third approach is similar to the second, taking the output of both classifiers as input, but this time using simple rules to determine a final result.

7.1 Combining features approach

The first and most obvious approach for combining the two classification methods is to train *one* classifier using both the linguistic- and visual features. There is however a big problem with this, and that is the exploding feature space. For the Linguistic and Context Based Classifiers we had to find an optimal feature subset out of 14 different features, while for the Visual Focus of Attention Classifier, there were 8 different features. To do this we had to do $2^{14} = 16.384$ experiments for the linguistic classifier and $2^8 = 256$ experiments for the visual classifier. If we where to use all features to train a classifier and still wish to find the optimal feature subset, there are $2^{22} = 4.194.304$ experiments to do for every classification method, which is simply too time consuming.

Therefore, we have to make some compromises. Because for both individual classifiers the MultilayerPerceptron scored very high, we only consider this classification method here. Furthermore, because doing over 4 million experiments would still take too long, we only use those features that were in the optimal feature subsets of the linguistic- and visual classifiers. These features are listed in Table 7.3.

The last column in Table 7.3 has a checkmark for every feature that is in the optimal feature subset for this classifier. The results of this classification can be seen in Table 7.4.

- Accuracy: 91,56
- Recall: 36,62
- Precision: 70,82
- F-Measure: 48,28

The result of this combination of features approach of 91,56% accuracy is better than the results of the two individual approaches. The total increase over the base score of 89.24% is **2.32%**. When we compare it to the hypothesized maximum of 92,47%, we are at 72% of what we can achieve.

Table 7.3: Features used and selected for the Combining Features Approach Classifier.

Paragraph	Feature	Selected
6.1.1	Type of the current Dialogue Act	✓
6.1.2	Short Dialogue Act	✓
6.1.3	Number of Words in the current Dialogue Act	✓
6.1.4	Contains 1st person singular Personal Pronoun	✓
6.1.6	Contains 2nd person singular/plural Personal Pronoun	✓
6.2.2	Type of Previous Dialogue Act	
6.2.4	Previous Dialogue Act addressed to me	
6.2.5	Activity History	✓
6.2.6	Previous Dialogue Act uttered by me	
6.2.7	Speaker Diversity History	✓
7.1.2	Total Time Everyone Looks at Me (Normalized)	✓
7.1.3	Total Time Speaker Looks at Me	✓
7.1.4	Total Time Speaker Looks at Me (Normalized)	✓
7.1.8	Number of Participants Looking At Me	

	Yes	No	Total
Yes	**182**	315	497
No	75	**4048**	4123
Total	257	4363	4620

Table 7.4: Confusion matrix for results of the Combining Features Approach.

7.2 Classification of results approach

For this approach we try to create a classifier that can predict the real class for a dialogue act 'yes' or 'no' (addressed to me, or not), based on the predictions of the visual- and linguistic classifiers. To do this we create a new corpus based on the predictions on the test set of both classifiers. The features for this corpus are defined as follows:

Actual Class - The actual class (yes/no) as taken from the corpus.

Predicted Class Linguistic Classifier - The decision of the Linguistic- and Context Based Classifier for this dialogue act (yes/no).

Predicted Class Visual Classifier - The decision of the Visual Focus of Attention Classifier for this dialogue act (yes/no).

Yes-probability Linguistic Classifier - Probability that class is 'yes' as predicted by the Linguistic and Context Based Classifier.

Yes-probability Visual Classifier - Probability that class is 'yes' as predicted by the Visual Focus of Attention Classifier.

The set of dialogue acts that are not used in the training sets of either classifiers, and that are annotated with visual focus of attention information, come from the meetings **ES2008a**, **IS1003b** and **IS1003d**. They total in 4620 dialogue act instances. This is split up in two-thirds for the training set (3080 instances) and one third for the test set (1540 instances). The class distribution for the test set is 89.87% *no* versus 10.13% *yes* (so the basescore is 89.87%). We have trained a total of 35 different classifiers on this training set and tested on the test set. The result achieved by the different classifiers can be seen in Figure 7.1.

Although the MultiLayer Perceptron classifier scores very high again (92.47%), the Ridor (or RIpple DOwn Rule learner) outperforms it a little bit (92.53%). The high performance of a rule based classifier makes sense for this sort of task, because the classifier has to make a decision based on two 'predictions'. Intuitively it translates to learning a rule that decides when the linguistic classifier is right, and when the focus of attention classifier is right. We will explore the rule-based approach some more in the next section (7.3). For now it is good to see that the results, 92.53% classified correctly are not only higher than those of the individual classifiers, but higher than the hypothesized maximum score of 92.47% from Section 4.1. On the scale of the base- and maximum scores, this result can be seen as 102% of what we can achieve. This may seem strange, but remember that the *hypothesized maximum* score should merely be seen as an estimated value with which to compare the classifier results in order to have a better sense of its meaning. Our score of 102% on this scale means that we *may be* reaching the limits of what our classifier can achieve, but this will be discussed in more detail in Chapter 9.

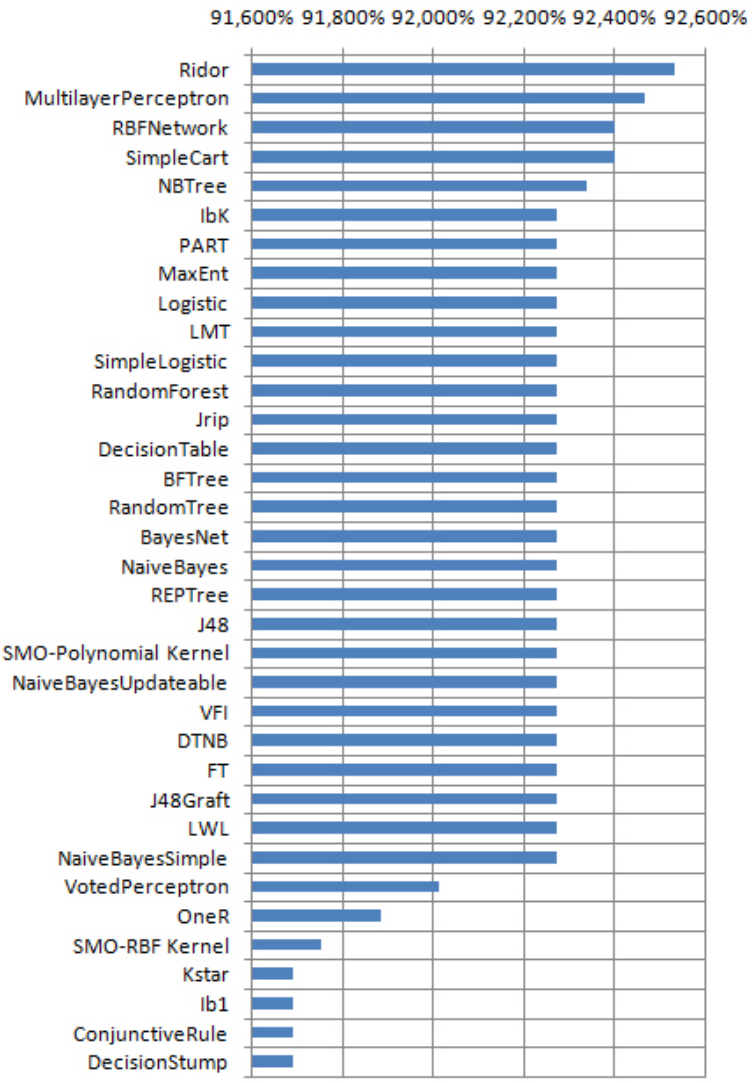


Figure 7.1: Classification results for 35 different classifiers using the results from the Linguistic- and VFOA Classifiers as input.

7.3 Simple rule-based approach

In this section we look at the results of the two individual classifiers, linguistic and visual, and see if we can find a simple rule to combine them to yield a good end result. First we will try some logical formulas, treating a classification of 'yes' as a 1, and 'no' as a 0. Afterwards we try weighing the probabilities as given by both classifiers, and try to find optimal values for the weights.

A first rule we will try is the logical AND, or: the predicted class is 'yes' if and only if both individual classifiers have predicted yes. Table 7.6 shows the results of the AND rule classifier.

	Yes	No	Total
Yes	**49**	448	497
No	3	**4120**	4123
Total	52	4568	4620

Table 7.5: Confusion matrix for results of logical AND rule classification.

- Accuracy: 90,24
- Recall: 9,86
- Precision: 94,23
- F-Measure: 17,85

As expected, this classification method has a very high precision, but very low recall, which unfortunately does not average out very well.

The second logical rule is then the OR rule: the predicted class is 'yes' if either or both of the individual classifiers predict a 'yes'. The results of this classifier can be seen in Table 7.6.

	Yes	No	Total
Yes	**234**	263	497
No	144	**3979**	4123
Total	378	4242	4620

Table 7.6: Confusion matrix for results of logical OR rule classification.

- Accuracy: 91,19
- Recall: 47,08
- Precision: 61,90
- F-Measure: 53,48

This rule already scores higher than the two classifiers individually, but unfortunately still not better than the more complex Rule Learning approach from the previous section (7.2). However, using only the class output of each of the classifiers, we can not do any better that this (see Table 7.7, this shows that the OR rule always makes the best decision).

VFOA Class	Linguistic Class	Yes	No
yes	yes	49	3
yes	no	89	63
no	yes	96	78
no	no	263	3979

Table 7.7: Individual Classifier results versus actual classes. The OR rule chooses *yes* for the first three rows, and *no* for the last, which is the best logical rule in this case.

7.4 Summary of results

We have tried four different ways of combining the results of the Linguistic- and Visual classifiers. The results of these are repeated in Table 7.8.

Method	Accuracy	% of Maximum
Combining Features	91.56	72%
Classification of Results	92.53	102 %
Logical AND Rule	90.24	31%
Logical OR Rule	91.19	60%

Table 7.8: Summary of results of the different approaches for combining classifiers.

From these figures you can see that the *Classification of Results* approach is by far the best for combining the classifiers. However, the first approach, of building a classifier using a combination of all features is not yet fully explored due to its computational difficulty. If the full feature set where to be explored, using a reasonable amount of different classification techniques, at least some improvements are to be expected. For now we must conclude that it's best to use the approach of *Classification of Results*, which achieved a score that exceeds the expectations set in our analysis of the data using inter-annotator agreement scores in Section 4.1. In the next Chapter we shall try to improve this score further by looking for other types of information.

Chapter 8

Using topic- and role information

Formal meetings are almost always governed by an agenda of sorts. The meeting is presumambly held for a reason; because there are certain points that need to be discussed. The use of agenda items gives structure to the meeting and helps to ensure that all the relevant points are being discussed, and satisfying outcomes will hopefully be achieved.

Up till now we have only used *local* information to predict the addressee of a speech action. We've analyzed the dialogue act itself, or some aspects of the previous one; and we've analyzed where the focus of attention of the meeting participants lies during or right before the speech. The availability of knowledge about the meeting structure, and the knowledge of participant's roles in the meeting, can help us on a higher level.

The study of meeting participant behavior in [18] has already shown that the activity of participants is highly dependent on the current *stage* of the meeting; during a presentation, the presenter is the one who is talking the most (assuming a polite audience), and an interface designer will probably be not as active as the financial adviser when the project budget is on the agenda. The knowledge of role and topic of discussion could intuitively also be harnassed to predict the addressee. Take the example of a presentation again; any question from the other meeting participants is more likely to be addressed to the presenter than to anyone else.

Given nothing but local information on a dialogue act, the prior probability that a dialogue act is *addressed to me* depends on the class-distribution only. In our previous examples this was around 89% of 'no' and 11% chance of 'yes'. In this chapter we try to use the knowledge of topic and role to make a more informed guess of this prior probability, by asking: what is the

chance of me being addressed, given that I have role R and topic T is being discussed.

Fortunately all the meetings in our train- and test corpora are annotated with role and topic information. In the AMI meetings, the participants have the task of designing a remote control, and each of the four members has either one of the following roles:

- User Interface specialist (UI).
- Marketing Expert (ME).
- Project Manager (PM).
- Industrial Designer (ID).

The topics that are used in the corpus range from organizational topics like 'opening' and 'closing' to more detailed contextual topics like 'user requirements'. The topics in the corpus are defined at the word level, so each word in the corpus has a corresponding topic in which it falls. But no dialogue acts are split up, so that they belong to two (or even more) topics at the same time.

Table 8.1 shows how often an *I-address* occurs during all the different topics that have been identified in the training corpus, for the four different participant roles. The **Topic** column shows all the topics that are extracted from the meetings in the training corpus. The second column (**Total**) is a count of the number of dialogue act instances that fall within this topic, sorted from most occuring topics to least. Between brackets are the total number of I-addressed utterances for that topic. The four columns that follow, count for each role (User Interface specialist, Marking Expert, Project Manager and Industrial Designer) how many of the dialogue act instances, *within that topic*, are addressed to me, having that particular role. For example, the corpus contains 3414 dialogue act instances that fall into the topic 'look and usability', 121 where addressed to the User Interface specialist, 151 to the Marketing Expert, 82 to the Project Manager, 112 to the Industrial Designer, and subsequently 2948 instances where not addressed to any one single indiviual.

Table 8.2 gives the overall distribution of topics and roles over the corpus. The first row contains all the information regarding dialogue acts in the topic 'look and usability'. The first column here is the total amount of these dialogue acts. The column UI then gives the amount of dialogue acts in the corpus, within that topic, where *my* role was User Interface specialist. Looking at the topic 'industrial designer presentation', you can see that the column with ID (Industrial Designer) has by far the lowest number in

that row. This is because the corpus does not contain dialogue acts that have been uttered by me (because those would never be addressed to me). During the industrial designer presentation, the industrial designer utters the most dialogue acts, so these have been left out. The number 138 is thus the number of dialogue acts that have been uttered by anyone but the ID.

Table 8.1: Distribution of I-Addressed Instances, per topic and role.

Topic	**Total (I)**	**UI**	**ME**	**PM**	**ID**
look and usability	3414 (466)	121	151	82	112
components, materials and energy sources	1881 (271)	69	32	56	114
costing	1311 (217)	26	52	98	41
evaluation of project process	1269 (180)	41	53	56	30
industrial designer presentation	1233 (124)	19	7	34	64
closing	1212 (169)	36	33	82	18
marketing expert presentation	1200 (124)	11	53	49	11
discussion	1194 (130)	36	15	37	42
agenda/equipment issues	999 (171)	42	16	76	37
drawing exercise	972 (181)	54	45	46	36
opening	906 (65)	8	9	38	10
interface specialist presentation	888 (87)	47	5	32	3
project specs and roles of participants	840 (149)	17	51	46	35
presentation of prototype(s)	774 (86)	28	12	25	21
project budget	765 (91)	22	21	20	28
evaluation of prototype(s)	762 (117)	21	58	25	13
other	714 (84)	13	12	30	29
chitchat	339 (46)	8	13	18	7
user requirements	324 (18)	5	8	3	2
new requirements	219 (28)	12	1	9	6
user target group	216 (50)	12	23	5	10
existing products	213 (9)	5	1	2	1
how to find when misplaced	204 (38)	17	14	7	0
trendwatching	24 (0)	0	0	0	0
Totals Per Topic:	21873	670	685	876	670

Table 8.2: Distribution of All Instances, per topic and role.

Topic	**Total**	**UI**	**ME**	**PM**	**ID**
look and usability	3414	822	798	935	859
components, materials and energy sources	1881	475	497	504	405
costing	1311	374	301	298	338
evaluation of project process	1269	361	264	296	348
industrial designer presentation	1233	360	396	339	138
closing	1212	339	342	177	354
marketing expert presentation	1200	366	129	327	378
discussion	1194	316	351	214	313
agenda/equipment issues	999	238	290	214	257
drawing exercise	972	245	237	223	267
opening	906	274	275	71	286
interface specialist presentation	888	81	282	239	286
project specs and roles of participants	840	243	225	133	239
presentation of prototype(s)	774	134	229	219	192
project budget	765	219	191	163	192
evaluation of prototype(s)	762	225	109	215	213
other	714	201	213	124	176
chitchat	339	96	67	81	95
user requirements	324	99	18	103	104
new requirements	219	59	73	20	67
user target group	216	57	33	63	63
existing products	213	18	69	61	65
how to find when misplaced	204	54	35	55	60
trendwatching	24	8	2	7	7
Totals Per Topic:	21873	5664	5426	5081	5702

Continuing the example of the first row in Table 8.1, we can now use a different set of a-priori probability of a dialogue act being *addressed to me*. We have **A** = Addressed to Me, **R** = Role and **T** = Topic. Then:

$$P(A) = \frac{TotalInstancesAddressedToMe}{TotalInstances}$$
$$P(A) = \frac{2901}{21873} = 0.132629$$

$$P(A|T = \text{`}lookandusability') = \frac{TotalInstancesAddressedToMeInTopic}{TotalInstancesInTopic}$$
$$P(A|T = \text{`}lookandusability') = \frac{466}{3414} = 0.136497$$

This probability hardly changes, because there is no reason to believe that I will be addressed more in a certain topic, regardless of my role in the meeting. But the probabilities already start to change when looking at the role alone:

$$P(A|R = \text{`}PM') = \frac{TotalInstancesAddressedToMeAsPM}{TotalInstancesAsPM}$$
$$P(A|R = \text{`}PM') = \frac{876}{5081} = 0.172407$$
$$P(A|R = \text{`}UI') = \frac{670}{5664} = 0.118291$$
$$P(A|R = \text{`}ME') = \frac{685}{5426} = 0.126244$$
$$P(A|R = \text{`}ID') = \frac{670}{5702} = 0.117503$$

You can see from this that when having the role of Industrial Designer, you are least likely to be addressed. But now let us look at this probability while at the topic of 'industrial designer presentation':

$$P(A|R = \text{`}ID' \wedge T = \text{`}IdPres.') = \frac{InstancesAddressedToMeAsID, duringIDPres}{TotalInstancesAsID, duringIDPres.}$$
$$P(A|R = \text{`}ID' \wedge T = \text{`}IdPres.') = \frac{64}{138} = 0.463768$$

This probability of 46% is much higher than the overal probability of being addressed of 13%. Now, we can calulate for every role and topic, the

probability of being addressed, and we have done so in Appendix B. We will try to use this role- and topic based a priori probability to improve upon the classification results that we already achieved in Chapter 7.

8.1 Classification using priors

Because the *Classification of Results Approach* of Chapter 7 proved to be the most succesful method in terms of final results, we will take this approach, and enhance it using our newly discoverd prior probabilities. Classification of Results was nothing more than a classic Machine Learning approach, so we have to translate our prior probability information into features. Remember that there were only *four* features used in the classification of results, so the feature space was very small. Therefore, there is no need to be economical when adding new features, so we added all that we could think of, which are the following four:

Topic - The topic in which the Dialogue Act was uttered.

Role - The role of 'me' for the classifier.

A-Priori - The exact prior probability that I am being addressed, as this role, in this topic, as taken from Appendix B.

A-Priori Nominal - The A-Priori value as a nominal feature: *low*, *normal* or *high* probability of being addressed. The borders of these three bins have been determined by looking at the distribution of chances in Appendix B. Most of them lie between 10% and 25%, so these are considered *normal*. A probability lower than 10% is considered *low* and a probability higher than 25% is *high*.

With these four new features added to the four from Section 7.1 we retrained all the different classifiers from those experiments. The results can be seen in Figure 8.1.

As you can see, the addition of the new features is cause for a big improvement for many classifiers, whereas others cannot use them at all. The best result, now achieved by the *Logistic Model Trees* (LMT) classifier, is 92,99%. This result, put on the scale of base- and maximum score, equals to 120%, which is by a fairly large margin better than the results of the inter-annotator agreement values. These results, among other issues, shall further be discussed in the next, concluding Chapter of this Thesis.

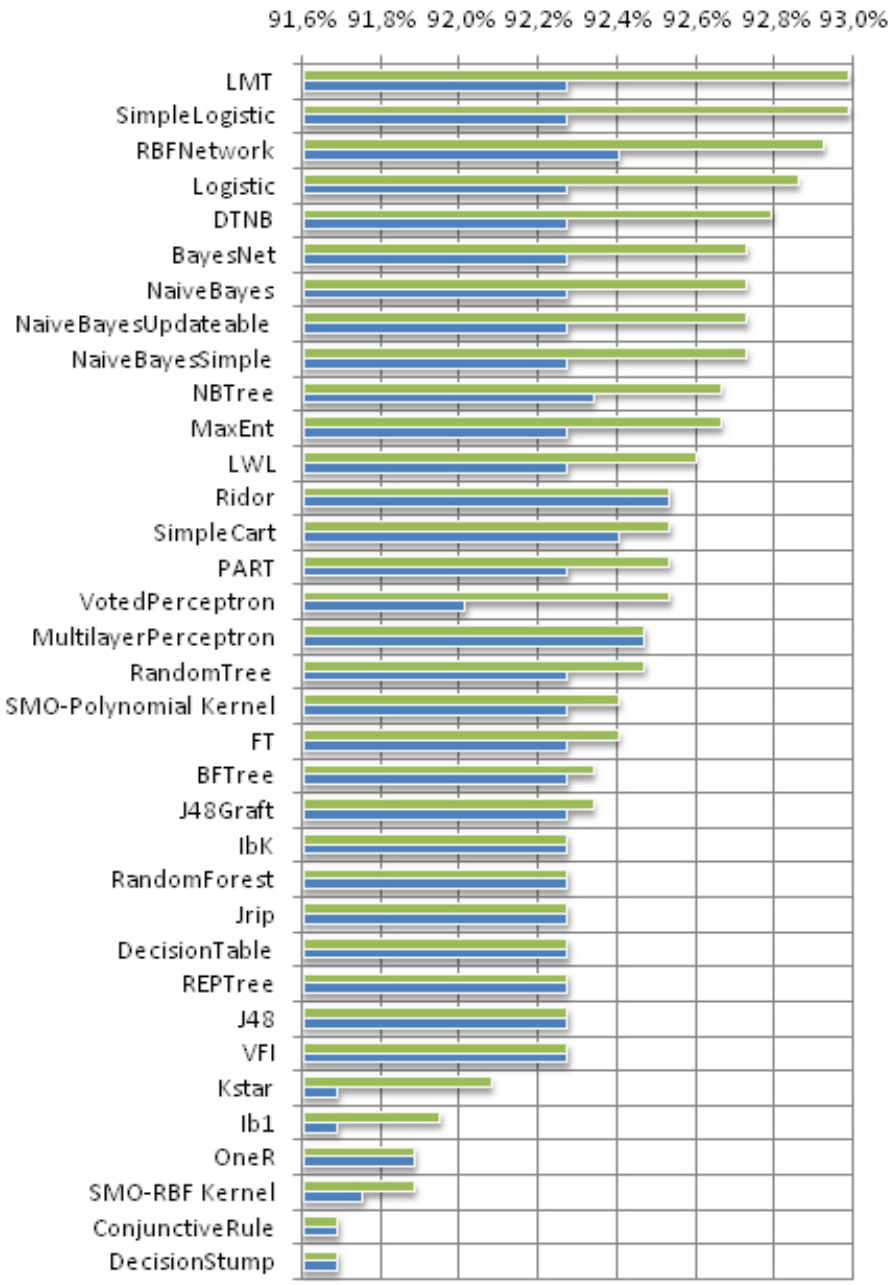


Figure 8.1: Classification results for 35 different classifiers. The blue, lower bars indicate the performance without using priors, the green, top bars indicate the performance using the topic- and role based prior probability information.

Chapter 9

Discussion

The work in this thesis presents a way of automatically determining the addressee of an utterance in a small group meeting that can be used in a real time application of a remote meeting assistant. This meeting assistant application is under development within the User Engagement and Floor Control Demonstrator project, which is a technology showcase demo of the AMIDA project, aimed at providing assistance to remote participants in hybrid meetings. In order for the detection module to function within the Demonstrator, we have strayed from the approach of many other works on addressee detection of answering the question "who is being addressed?". Instead we take the role of an assisting agent for a remote meeting participant, and try to answer the question "am I being addressed?". This is a fundamentally different approach for a number of reasons. First, we do not assume a fixed number of participants in the meeting. When you want to know if an utterance is addressed to the Group as a whole, or a certain individual in the group, you have to label each individual, and use these as class labels for the classifier. The classes in our case are either 'yes' or 'no', regardless of the number of participants in the meeting. Second, instead of having to distinguish between, for example, 5 classes (as is the case in [15]), we only need to seperate two. This intuitively makes the task easier, but it also makes it difficult to compare results with other work in the field.

The creation of the automatic addressee detection module was divided into four steps: 1) a classifier that uses linguistic and context information, 2) a classifier that uses visual focus of attention information, 3) the combination of these two classifiers, and 4) an enhancement of the classifier using topic and role information. We shall shortly discuss each step in the process before coming to the results afterwards.

The Linguistic- and Context Based Classifier of Chapter 5 uses all of the features that can be derived from the word- and dialogue act layers

of information in the corpus that have been used in earlier work like [15]. Additionally, we have modelled the context preceding the current dialogue act in terms of how active I was, how often I was addressed and how many different participants has spoken in a certain backwards looking window. We then trained 15 different classifiers using either the gold standard data for features based on addressing information, or a simulated online version where the output of the classifier of dialogue act DA_x is used as feature for dialogue act DA_{x+1}. The results of 90.93% accuracy, are, as expected, not the best we hope to achieve: only 53% on the scale of our base- and hypothesized maximum scores. There are however some improvements possible. The Dialogue Act type features for example, use a tagset of 7 tags, which is defined in [15]. This specific tagset may not be optimal for our setting, so improvement can be achieved by finding an optimal set. In a similar way, performance could slightly improve by adjusting the "number of words in the current dialogue act" feature, to use an optimal nominal distribution. The features that indiciate wheter the utterance contains a certain Personal Pronoun are based on the intuitive notion that certain Personal Pronouns like "we" or "our" could, for example, indicate a Group-address. But a thorough research on the appearance of all the words in the corpus related to addressees would be needed to optimize these kinds of features. Feature optimization aside, a larger improvement in results is expected when simply more data is available. Only 18 out of the 138 scenario meetings in the AMI corpus (13%) have been annotated with addressing information, but all of these have been with word- and dialogue act information. It is therefore not an unrealistic task to get more data, and it could greatly improve the performance of the classifiers.

The second step in the process of our work has been the Visual Focus of Attention Classifier of Chapter 6. It uses only the information from the focus of attention layer in the AMI corpus. Where lies the focus of every participant during the utterance, or right before the utterance? This information has been studied in much detail to derive a set of features that are as information dense as possible for the task of addressee detection. Because it still only resulted in 8 different features, we were able to train twice as much classifiers (30) as used for the Linguistic Classifier. Even though there were twice as much candidate classifiers, the results were on par with those ff the Linguistic ones: 90.8% accuracy, which, put on the scale of the base- and maximum scores amounts to 48% achieved. The fact that similar results are achieved while the amount of training data for the Visual Classifier is a third less than that for the Linguistic classifier, shows the high potential of looking at the visual cues. In terms of improvement, the amount of training data is the first thing that comes to mind. We have been rigorous in defining and selecting the features for the classifier, so there may not be much

improvement to be gained there.

What we were interested in mostly is of course the effect of combining the Linguistic and Visual features. Chapter 7 describes three different ways of doing that. Training classifiers with a combination of the 14 Linguistic and 8 Visual features proved difficult because of the large feature space. Therefore, this approach was not fully explored, which may explain the slightly dissapointing results: 91,56% accuracy (72% of expected maximum). Given more time, this method can be fully explored, but it remains hard to predict how much increase in performance can be expected. Using the output of the two individual classifiers in a simple OR rule (if either one of the classifiers thinks 'yes', the outcome is 'yes') resulted in slightly worse results (91,19% accuracy, 60% of expected maximum), but it is also just a simplistic method. Using the output of the two classifiers for training a third classifier then proved to be the most succesful method. With a result of 92,53% accuracy, the expected maximum score of 92,47% is surpassed by 2%.

As explained in Chapter 4, it is hard to say whether these results that are slightly over the hypothesized maximum, are indeed the best we can hope to achieve. We do know that the selection of the right addressee for an utterance is an inherently ambiguous task, and that our classifier could fit right in with the annotators, when comparing their results on annotating a meeting. For me, this is at least an indication that these results can be considered good.

In Chapter 8, the final chapter, we try to increase our performance by using two previously untapped sources of information: topics and roles. We calculated from the training set of the corpus, what the *a priori* probabilities are of being addressed given the fact that you have a certain role in the meeting, and that a certain topic is being discussed. We then derived features from this, and re-trained the *Classification of Results* classifiers. The results are very positive, with now 92.99% accuracy achieved (120% of the hypothesized maximum score). The reason why we treat this enhancement seperately is because this information can not be used in the intended User Engagement and Floor Control Demonstrator. First you need to know what the roles of the meeting participants are. This is only a minor problem ofcourse; you can imagine simply entering this information into the system at the start of a meeting. But there are two real problems. First, there also needs to be an automatic topic detection module, and this is, altough not impossible, a lot more difficult. The module would have to keep track of the conversation and look for cues from a discussion leader as to when he announces a new topic. It then also needs to "classify" this topic using either a

predefined set of topics, or by creating such a set as it processes more meetings. Then, the a priori probabilities could be calculated using the output of the addressing system. The problem of course is that the addressing module is not always right, and the information becomes imperfect. The use of topic- and role information can thus potentially greatly enhance the performance of an addressing detection module, but it needs to be trained on that specific set of topics and roles that are needed for the intended application.

The intended application for the classifiers that we've build here is the support of remote participants in hybrid meetings; meetings where some of the participants are gathered around a table locally, and one or more are joined via teleconferencing software. The data that is used here is exclusively gathered from "normal", local meetings. The difference between these two types of meetings are apparent, and so, a decrease in performance is expected when using our software in the hybrid meeting setting. The creation of richly annotated corpora of hybrid meetings is needed to ensure the quality of software that has been tested with local meetings, but designed for the hybrid setting.

The work in this thesis shows how automatic addressee detection can in theory be done for an online application. The results are on par with what can be expected to be achieved for this type of setting. The impact of the decline in performance when switching to the hybrid meeting setting can not be accurately foreseen. Further research should focus on testing the application in real meetings. Only then can we be sure whether the achieved accuracy figures of our classifiers are indeed any good...

Acknowledgements

First of all I would like to thank my Master Thesis committee members Dirk Heylen, Betsy van Dijk and Dennis Hofs for their guidance and comments on the various early versions of this work. Of course I thank my parents Rieks op den Akker and Marion Geerdes for giving me the opportunity for following this study at the University of Twente. I thank Hendri Hondorp for his technical assistance in working on the UEFC meeting room. I wish to thank Ariane for her mental support and loving care. I also thank all my friends and co-students at the university for a very good time throughout my university years. And finally a special thanks to my father Rieks, for his guidance and wisdom throughout my scientific endeavours...

Bibliography

[1] E. Goffman, Forms of Talk. Philadelphia, Pennsylvania 19104-4011: University of Pennsylvania Press, 1981.

[2] D. Traum, "Issues in multiparty dialogues," in Advances in Agent Communication, pp. 201–211, 2004.

[3] H. Clark, Using language. The Edinburgh Building, Cambridge CB2 2RU, UK: Cambridge University Press, 1996.

[4] N. Jovanovic and R. Op den Akker, "Towards automatic addressee identification in multi-party dialogues," in Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue, (Cambridge, Massachusetts, USA), pp. 89–92, Association for Computational Linguistics, 2004.

[5] H. G. Lerner, "Selecting next speaker: The context-sensitive operation of a context-free organization," in Language in Society, vol. 32, pp. 177–201, 2003.

[6] M. Argyle, R. Ingham, F. Alkema, and M. McCallin, "The different functions of gaze," in Semiotica, pp. 10–32, 1973.

[7] P. M. Aoki, S. M. H., L. Plurkowski, and J. D. Thornton, "Where's the "party" in "multi-party"? analyzing the structure of small-group sociable talk," in Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work, pp. 393–402, 2006.

[8] Y. Takemae and S. Ozawa, "Automatic addressee identification based on participants' head orientation and utterances for multiparty conversations," in Multimedia and Expo, 2006 IEEE International Conference on, pp. 1285–1288, 2006.

[9] A. Jaimes, K. Omura, T. Nagamine, and K. Hirata, "Memory cues for meeting video retrieval," in Proceedings of the the 1st ACM workshop on Continuous archival and retrieval of personal experiences, pp. 74–85, 2004.

[10] D. Lalanne, A. Lisowska, E. Bruno, M. Flynn, M. Georgescul, M. Guillemot, B. Janvier, S. Marchand-Maillet, M. Melichar, N. Moenne-Loccoz, A. Popescu-Belis, M. Rajman, M. Rigamonti, D. von Rotz, and P. Wellner, "The im2 multimodal meeting browser family," tech. rep., European Foundation for the Improvement of Living and Working Conditions, 2005.

[11] H. Op den Akker, "Question answering as meeting browser interface," in Proceedings of the 4th Twente Student Conference on Information Technology, pp. 157–164, 2006.

[12] Y. Takemae, K. Otsuka, and M. Naoki, "Impact of video editing based on participants' gaze in multiparty conversation," in Proceedings of the Conference on Human Factors in Computing Systems CHI '04, pp. 1333–1336, 2004.

[13] S. Castronovo, J. Frey, and P. Poller, "A generic layout-tool for summaries of meetings in a constraint-based approach," in Lecture Notes in Computer Science - Proceedings of the 5th International Workshop on Machine Learning for Multimodal Interaction, vol. 5237, pp. 248–259, 2008.

[14] M. D. Katzenmaier, Identifying the Addressee in Human-Human-Robot Interactions based on Head Pose and Speech. PhD thesis, Universitat Karlsruhe ISL, June 2004.

[15] N. Jovanovic, To whom it may concern : adressee identification in face-to-face meetings. PhD thesis, University of Twente, 2007.

[16] T. de Zeeuw, "A rule based addressee identification algorithm used on the ami corpus," in Proceedings of the 8th Twente Student Conference on IT, 2008.

[17] R. Op den Akker and M. Theune, "How do i address you? - modelling addressing behavior based on an analysis of a multi-modal corpus of conversational discourse," in Proceedings of the AISB 2008 Symposium on Multimodal Output Generation (MOG 2008), Aberdeen, UK, pp. 10–17, 2008.

[18] "Amida deliverable d1.3: Qualitative analysis of interactions in face to face and remote meetings," tech. rep., University of Twente, 2008.

[19] K. S. Suh, "Impact of communication medium on task performance and satisfaction: an examination of media-richness theory," in Information and Management, vol. 35, pp. 295–312, 1999.

[20] S. Whittaker, "Theories and methods in mediated communication," in The Handbook of Discourse Processes, pp. 243–286, Erlbaum, 2003.

[21] R. Op den Akker, D. Hofs, H. Hondorp, H. Op den Akker, J. Zwiers, and A. Nijholt, "Engagement and floor control in hybrid meetings." Submitted for COST 2009, 2009.

[22] A. Popescu-Belis, E. Boertjes, K. Jonathan, P. Poller, S. Castronovo, T. Wilson, A. Jaimes, and J. Carletta, "The amida content linking device: Just-in-time document retrieval in meetings," in Lecture Notes in Computer Science - Proceedings of the 5th International Workshop on Machine Learning for Multimodal Interaction, vol. 5237, pp. 272–283, 2008.

[23] T. Hain, A. El Hannani, S. N. Wrigley, and V. Wan, "Automatic speech recognition for scientific purposes - webasr," in Proceedings of the international conference on spoken language processing (Interspeech 2008), 2008.

[24] H. Op den Akker and C. Schulz, "Exploring features and classifiers for dialogue act segmentation," in Lecture Notes in Computer Science - Proceedings of the 5th International Workshop on Machine Learning for Multimodal Interaction, vol. 5237, pp. 196–207, 2008.

[25] S. Germesin, T. Becker, and P. Poller, "Determining latency for on-line dialog act classification," in Poster Session for the 5th International Workshop on Machine Learning for Multimodal Interaction, vol. 5237, 2008.

[26] S. Ba and J.-M. Odobez, "Recognizing human visual focus of attention from head pose in meetings," in IEEE Transaction on Systems, Man, and Cybernetics, Part B (Trans. SMC-B), vol. 39, pp. 16–33, 2009.

[27] N. Jovanovic, R. Op den Akker, and A. Nijholt, "A corpus for studying addressing behaviour in multi-party dialogues," in Language Resources and Evaluation, pp. 5–23, Springer Netherlands, 2006.

[28] E. Alpaydin, Introduction to Machine Learning. 5 Cambridge Center, Cambridge, MA 02142: The MIT Press, 2004.

[29] I. McCowan, J. Carletta, W. Kraaij, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, W. Post, D. Reidsma, and P. Wellner, "The ami meeting corpus," in Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research, 2005.

[30] J. Cohen, "A coefficient of agreement for nominal scales," Educational and Psychological Measurement, vol. 20, pp. 37–46, April 1960.

[31] D. Reidsma, Annotations and Subjective Machines. PhD thesis, University of Twente, 2004.

[32] I. H. Witten and E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, 1st ed., October 1999.

[33] G. Cooper and E. Herskovits, "A bayesian method for the induction of probabilistic networks from data," Machine Learning, vol. 9, no. 4, pp. 309–347, 1992.

[34] H. Shi, "Best-first decision tree learning," Master's thesis, University of Waikato, Hamilton, NZ, 2007. COMP594.

[35] R. Kohavi, "The power of decision tables," in 8th European Conference on Machine Learning, pp. 174–189, Springer, 1995.

[36] M. Hall and E. Frank, "Combining naive bayes and decision tables," in Proceedings of the 21st Florida Artificial Intelligence Society Conference (FLAIRS), AAAI press, 2008.

[37] J. Gama, "Functional trees," vol. 55, no. 3, pp. 219–250, 2004.

[38] D. Aha and D. Kibler, "Instance-based learning algorithms," Machine Learning, vol. 6, pp. 37–66, 1991.

[39] R. Quinlan, C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann Publishers, 1993.

[40] G. Webb, "Decision tree grafting from the all-tests-but-one partition," (San Francisco, CA), Morgan Kaufmann, 1999.

[41] W. W. Cohen, "Fast effective rule induction," in Twelfth International Conference on Machine Learning, pp. 115–123, Morgan Kaufmann, 1995.

[42] J. G. Cleary and L. E. Trigg, "K*: An instance-based learner using an entropic distance measure," in 12th International Conference on Machine Learning, pp. 108–114, 1995.

[43] N. Landwehr, M. Hall, and E. Frank, "Logistic model trees," Machine Learning, vol. 95, no. 1-2, pp. 161–205, 2005.

[44] S. le Cessie and J. van Houwelingen, "Ridge estimators in logistic regression," Applied Statistics, vol. 41, no. 1, pp. 191–201, 1992.

[45] C. Atkeson, A. Moore, and S. Schaal, "Locally weighted learning," AI Review, 1996.

[46] G. H. John and P. Langley, "Estimating continuous distributions in bayesian classifiers," in Eleventh Conference on Uncertainty in Artificial Intelligence, (San Mateo), pp. 338–345, Morgan Kaufmann, 1995.

[47] R. Duda and P. Hart, Pattern Classification and Scene Analysis. New York: Wiley, 1973.

[48] R. Kohavi, "Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid," in Second International Conference on Knoledge Discovery and Data Mining, pp. 202–207, 1996.

[49] B. Martin, "Instance-based learning: Nearest neighbor with generalization," Master's thesis, University of Waikato, Hamilton, New Zealand, 1995.

[50] R. Holte, "Very simple classification rules perform well on most commonly used datasets," Machine Learning, vol. 11, pp. 63–91, 1993.

[51] E. Frank and I. H. Witten, "Generating accurate rule sets without global optimization," in Fifteenth International Conference on Machine Learning, pp. 144–151, Morgan Kaufmann, 1998.

[52] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[53] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees. Belmont, California: Wadsworth International Group, 1984.

[54] G. Demiroz and A. Guvenir, "Classification by voting feature intervals," in 9th European Conference on Machine Learning, pp. 85–92, Springer, 1997.

[55] Y. Freund and R. E. Schapire, "Large margin classification using the perceptron algorithm," in 11th Annual Conference on Computational Learning Theory, (New York, NY), pp. 209–217, ACM Press, 1998.

Appendix A

Inter-annotator confusion matrices

This appendix contains the inter annotator confusion data on the addressee annotations of meeting IS1003d that is used in Section 4.1. There are four different annotators that have annotated the meeting: **s9553330**, **vkaraisk**, **marisa** and **dharsi**, so there are 6 distinct annotator pairs. For each of those pairs, the confusion matrices for all class labels is given first[1]. Then, the confusion matrices from the persective of the individual meeting participants, A, B, C and D are given. This means that the annotations are considered to have been annotated as '*addressed to A*' or '*not addressed to A*'. For each of these individual meeting participant cases, recall, precision, f-measure and accuracy values are given. Then, the data for each four individual cases is combined, to calculate the total values for that annotator pair.

A.1 s9553330 and vkaraisk

	A	B	C	D	Group	Total
A	**29**				10	39
B		**14**			8	22
C			**32**		7	39
D	1		1	**49**	18	69
Group	21	10	19	22	**171**	243
Total	51	24	52	71	214	412

Table A.1: Confusion matrix for pair s9553330 and vkaraisk. Alpha Krippendorff: 0.55, Kappa Cohen: 0.55.

[1]Note that *Unknown* labels are treated as *Group* here.

Participant A

	A	$\neg A$	Total
A	**29**	10	39
$\neg A$	22	**351**	373
Total	51	361	412

Table A.2: Confusion matrix for pair s955330 and vkaraisk, considering addressed to A or not.

- Accuracy: 92,23
- Recall: 74,36
- Precision: 56,86
- F-Measure: 64,44

Participant B

	B	$\neg B$	Total
B	**14**	8	22
$\neg B$	10	**380**	390
Total	24	388	412

Table A.3: Confusion matrix for pair s955330 and vkaraisk, considering addressed to B or not.

- Accuracy: 95,63
- Recall: 63,64
- Precision: 58,33
- F-Measure: 60,87

Participant C

	C	$\neg C$	Total
C	**32**	7	39
$\neg C$	20	**353**	373
Total	52	360	412

Table A.4: Confusion matrix for pair s955330 and vkaraisk, considering addressed to C or not.

- Accuracy: 93,45
- Recall: 82,05
- Precision: 61,54
- F-Measure: 70,33

Participant D

	D	$\neg D$	Total
D	**49**	20	69
$\neg D$	22	**321**	343
Total	71	341	412

Table A.5: Confusion matrix for pair s955330 and vkaraisk, considering addressed to D or not.

- Accuracy: 89,81
- Recall: 71,01
- Precision: 69,01
- F-Measure: 70,00

All Participants Combined

	Yes	No	Total
Yes	**124**	45	169
No	74	**1405**	1479
Total	198	1450	1648

Table A.6: Confusion matrix for pair s955330 and vkaraisk, considering addressed to *me* or not.

- Accuracy: 92,78
- Recall: 73,37
- Precision: 62,63
- F-Measure: 67,58

A.2 marisa and s9553330

	A	B	C	D	Group	Total
A	**34**			2	30	66
B		**21**	1		16	38
C			**28**		23	51
D		1		**61**	24	86
Group	15	5	4	32	**186**	242
Total	49	27	33	95	279	483

Table A.7: Confusion matrix for pair marisa and s9553330. Alpha Krippendorff: 0.51, Kappa Cohen: 0.51.

Participant A

	A	$\neg A$	Total
A	**34**	32	66
$\neg A$	15	**402**	417
Total	49	434	483

Table A.8: Confusion matrix for pair marisa and s9553330, considering addressed to A or not.

- Accuracy: 90,27
- Recall: 51,52
- Precision: 69,39
- F-Measure: 59,13

Participant B

	B	$\neg B$	Total
B	**21**	17	38
$\neg B$	6	**439**	445
Total	27	456	483

Table A.9: Confusion matrix for pair marisa and s9553330, considering addressed to B or not.

- Accuracy: 95,24
- Recall: 55,26
- Precision: 77,78
- F-Measure: 64,62

Participant C

	C	$\neg C$	Total
C	**28**	23	51
$\neg C$	5	**427**	432
Total	33	450	483

Table A.10: Confusion matrix for pair marisa and s9553330, considering addressed to C or not.

- Accuracy: 94,20
- Recall: 54,90
- Precision: 84,85
- F-Measure: 66,67

Participant D

	D	$\neg D$	Total
D	**61**	25	86
$\neg D$	34	**363**	397
Total	95	388	483

Table A.11: Confusion matrix for pair marisa and s9553330, considering addressed to D or not.

- Accuracy: 87,78
- Recall: 70,93
- Precision: 64,21
- F-Measure: 67,40

All Participants Combined

	Yes	No	Total
Yes	**144**	97	241
No	60	**1631**	1691
Total	204	1728	1932

Table A.12: Confusion matrix for pair marisa and s9553330, considering addressed to *me* or not.

- Accuracy: 91,87
- Recall: 59,75
- Precision: 70,59
- F-Measure: 64,72

A.3 marisa and vkaraisk

	A	B	C	D	Group	Total
A	**46**				28	74
B	1	**25**			13	39
C			**38**	1	11	50
D				**63**	20	83
Group	23	6	13	14	**177**	233
Total	70	31	51	78	249	479

Table A.13: Confusion matrix for pair marisa and vkaraisk. Alpha Krippendorff: 0.60, Kappa Cohen: 0.60.

Participant A

	A	$\neg A$	Total
A	**46**	28	74
$\neg A$	24	**381**	405
Total	70	409	479

Table A.14: Confusion matrix for pair marisa and vkaraisk, considering addressed to A or not.

- Accuracy: 89,14
- Recall: 62,16
- Precision: 65,71
- F-Measure: 63,89

Participant B

	B	$\neg B$	Total
B	**25**	14	39
$\neg B$	6	**434**	440
Total	31	448	479

Table A.15: Confusion matrix for pair marisa and vkaraisk, considering addressed to B or not.

- Accuracy: 95,82
- Recall: 64,10
- Precision: 80,65
- F-Measure: 71,43

Participant C

	C	$\neg C$	Total
C	**38**	12	50
$\neg C$	13	**416**	429
Total	51	428	479

Table A.16: Confusion matrix for pair marisa and vkaraisk, considering addressed to C or not.

- Accuracy: 94,78
- Recall: 76,00
- Precision: 74,51
- F-Measure: 75,25

Participant D

	D	$\neg D$	Total
D	**63**	20	83
$\neg D$	15	**381**	396
Total	78	401	479

Table A.17: Confusion matrix for pair marisa and vkaraisk, considering addressed to D or not.

- Accuracy: 92,69
- Recall: 75,90
- Precision: 80,77
- F-Measure: 78,26

All Participants Combined

	Yes	No	Total
Yes	**172**	74	246
No	58	**1612**	1670
Total	230	1686	1916

Table A.18: Confusion matrix for pair marisa and vkaraisk, considering addressed to *me* or not.

- Accuracy: 93,11
- Recall: 69,92
- Precision: 74,78
- F-Measure: 72,27

A.4 marisa and dharshi

	A	B	C	D	Group	Total
A	**14**		1		31	46
B		**10**	1		20	31
C			**31**		21	52
D				**16**	43	59
Group	6	2	3	3	**203**	217
Total	20	12	36	19	318	405

Table A.19: Confusion matrix for pair marisa and dharshi. Alpha Krippendorff: 0.39, Kappa Cohen: 0.42.

Participant A

	A	$\neg A$	Total
A	**14**	32	46
$\neg A$	6	**353**	359
Total	20	385	405

Table A.20: Confusion matrix for pair marisa and dharshi, considering addressed to A or not.

- Accuracy: 90,62
- Recall: 30,43
- Precision: 70,00
- F-Measure: 42,42

Participant B

	B	$\neg B$	Total
B	**10**	21	31
$\neg B$	2	**372**	374
Total	12	393	405

Table A.21: Confusion matrix for pair marisa and dharshi, considering addressed to B or not.

- Accuracy: 94,32
- Recall: 32,26
- Precision: 83,33
- F-Measure: 46,51

Participant C

	C	$\neg C$	Total
C	**31**	21	52
$\neg C$	5	**348**	353
Total	36	369	405

Table A.22: Confusion matrix for pair marisa and dharshi, considering addressed to C or not.

- Accuracy: 93,58
- Recall: 59,62
- Precision: 86,11
- F-Measure: 70,45

Participant D

	D	$\neg D$	Total
D	**16**	43	59
$\neg D$	3	**343**	346
Total	19	386	405

Table A.23: Confusion matrix for pair marisa and dharshi, considering addressed to D or not.

- Accuracy: 88,64
- Recall: 27,12
- Precision: 84,21
- F-Measure: 41,03

All Participants Combined

	Yes	No	Total
Yes	**71**	117	188
No	16	**1416**	1432
Total	87	1533	1620

Table A.24: Confusion matrix for pair marisa and dharshi, considering addressed to D or not.

- Accuracy: 91,79
- Recall: 37,77
- Precision: 81,61
- F-Measure: 51,64

A.5 vkaraisk and dharshi

	A	B	C	D	Group	Total
A	**19**				29	48
B		**8**			8	16
C			**25**	1	15	41
D			1	**14**	37	52
Group	2	2	6	4	**173**	187
Total	21	10	32	19	262	344

Table A.25: Confusion matrix for pair vkaraisk and dharshi. Alpha Krippendorff: 0.44, Kappa Cohen: 0.45.

Participant A

	A	$\neg A$	Total
A	**19**	29	48
$\neg A$	2	**294**	296
Total	21	323	344

Table A.26: Confusion matrix for pair vkaraisk and dharshi, considering addressed to A or not.

- Accuracy: 90,99
- Recall: 39,58
- Precision: 90,48
- F-Measure: 55,07

Participant B

	B	$\neg B$	Total
B	**8**	8	16
$\neg B$	2	**326**	328
Total	10	334	344

Table A.27: Confusion matrix for pair vkaraisk and dharshi, considering addressed to B or not.

- Accuracy: 97,09
- Recall: 50,00
- Precision: 80,00
- F-Measure: 61,54

Participant C

	C	$\neg C$	Total
C	**25**	16	41
$\neg C$	7	**296**	303
Total	32	312	344

Table A.28: Confusion matrix for pair vkaraisk and dharshi, considering addressed to C or not.

- Accuracy: 93,31
- Recall: 60,98
- Precision: 78,12
- F-Measure: 68,49

Participant D

	D	$\neg D$	Total
D	**14**	38	52
$\neg D$	5	**287**	292
Total	19	325	344

Table A.29: Confusion matrix for pair vkaraisk and dharshi, considering addressed to D or not.

- Accuracy: 87,50
- Recall: 26,92
- Precision: 73,68
- F-Measure: 39,44

All Participants Combined

	Yes	No	Total
Yes	**66**	91	157
No	16	**1203**	1219
Total	82	1294	1376

Table A.30: Confusion matrix for pair vkaraisk and dharshi, considering addressed to *me* or not.

- Accuracy: 92,22
- Recall: 42,04
- Precision: 80,49
- F-Measure: 55,23

A.6 s9553330 and dharshi

	A	B	C	D	Group	Total
A	**18**				27	45
B		**8**			8	16
C			**30**		8	38
D				**20**	55	75
Group	7	3	7	5	**234**	256
Total	25	11	37	25	332	430

Table A.31: Confusion matrix for pair s9553330 and dharshi. Alpha Krippendorff: 0.45, Kappa Cohen: 0.46.

Participant A

	A	$\neg A$	Total
A	**18**	27	45
$\neg A$	7	**378**	385
Total	25	405	430

Table A.32: Confusion matrix for pair s9553330 and dharshi, considering addressed to A or not.

- Accuracy: 92,09
- Recall: 40,00
- Precision: 72,00
- F-Measure: 51,43

Participant B

	B	$\neg B$	Total
B	**8**	8	16
$\neg B$	3	**411**	414
Total	11	419	430

Table A.33: Confusion matrix for pair s9553330 and dharshi, considering addressed to B or not.

- Accuracy: 97,44
- Recall: 50,00
- Precision: 72,73
- F-Measure: 59,26

Participant C

	C	$\neg C$	Total
C	**30**	8	38
$\neg C$	7	**385**	392
Total	37	393	430

Table A.34: Confusion matrix for pair s9553330 and dharshi, considering addressed to C or not.

- Accuracy: 96,51
- Recall: 78,95
- Precision: 81,08
- F-Measure: 80,00

Participant D

	D	$\neg D$	Total
D	**20**	55	75
$\neg D$	5	**350**	355
Total	25	405	430

Table A.35: Confusion matrix for pair s9553330 and dharshi, considering addressed to D or not.

- Accuracy: 86,05
- Recall: 26,67
- Precision: 80,00
- F-Measure: 40,00

All Participants Combined

	Yes	No	Total
Yes	**76**	98	174
No	22	**1524**	1546
Total	98	1622	1720

Table A.36: Confusion matrix for pair s9553330 and dharshi, considering addressed to *me* or not.

- Accuracy: 93,02
- Recall: 43,68
- Precision: 77,55
- F-Measure: 55,88

Appendix B

Prior probability distribution

Table B.1 lists the prior probabilities of being addressed for every combination of topic and role. These probabilities are used in Section 8.1. The column '**Total**' contains the probabilities of being addressed while that topic is being discussed, regardless of role, while the totals in the last row are the probabilities of being addressed when having a certain role, regardless of topic. The value in the last row, column 'Total' is the overall average probability of being addressed regardless of role or topic.

Table B.1: Distribution of Prior Probabilities per topic and role.

Topic	**Total**	**UI**	**ME**	**PM**	**ID**
look and usability	13.6%	14.7%	18.9%	8.7%	13.0%
comp., mat. and energy sources	14.4%	14.5%	6.4%	11.1%	28.2%
costing	16.6%	7.0%	17.3%	32.9%	12.1%
evaluation of project process	14.2%	11.4%	20.1%	18.9%	8.6%
ID presentation	10.1%	5.3%	1.8%	10.0%	46.4%
closing	13.9%	10.6%	9.7%	46.3%	5.1%
ME presentation	10.3%	3.0%	41.1%	15.0%	2.9%
discussion	10.9%	11.4%	4.3%	17.3%	13.4%
agenda/equipment issues	17.1%	17.7%	5.5%	35.5%	14.4%
drawing exercise	18.6%	22.0%	19.0%	20.6%	13.5%
opening	7.2%	2.9%	3.3%	53.5%	3.5%
UI presentation	9.8%	58.0%	1.8%	13.4%	1.1%
pr. specs & roles	17.7%	7.0%	22.7%	34.6%	14.6%
presentation of prototype(s)	11.1%	20.9%	5.2%	11.4%	10.9%
project budget	11.9%	10.0%	11.0%	12.3%	14.6%
evaluation of prototype(s)	15.4%	9.3%	53.2%	11.6%	6.1%
other	11.8%	6.5%	5.6%	24.2%	16.5%
chitchat	13.6%	8.3%	19.4%	22.2%	7.4%
user requirements	5.6%	5.0%	44.4%	2.9%	1.9%
new requirements	12.8%	20.3%	1.4%	45.0%	9.0%
user target group	23.2%	21.0%	69.7%	7.9%	15.9%
existing products	4.2%	27.8%	1.5%	3.3%	1.5%
how to find when misplaced	18.6%	31.5%	40.0%	12.7%	0.0%
trendwatching	0.0%	0.0%	0.0%	0.0%	0.0%
Totals Per Topic:	13.3%	11.8%	12.6%	17.2%	11.8%

Appendix C

Description of classifiers

This appendix gives a short description of all the different machine classifier names that are mentioned in this Thesis. The descriptions, as well as the citations, are taken from the WEKA documentation. For more information on many of these classification technique, see the book that accompanies the WEKA toolkit: [32].

BayesNet Bayesian Network classifier using a K2 search algorithm (a hill climbing algorithm restricted by an order on the variables, see [33]).

BFTree Best First decision tree classifier. For more information see [34].

ConjunctiveRule Conjunctive Rule Learner; creates rules consisting of antecedents "AND"ed together (IF P AND Q THEN CLASS-X). Antecedents are selected by computings its Information Gain, and then pruned using Reduced Error Pruning (REP) or simple pre-pruning, based in the number of antecedents.

DecisionStump A Decision Stump classifier is a decision tree with only one level.

DecisionTable A Decision table majority classifier. For more information see: [35].

DTNB Decision Table / Naive Bayes hybrid classifier. At each point in the search, the algorithm evaluates the merit of dividing the attributes into two disjoint subsets: one for the decision table, the other for naive Bayes. A forward selection search is used, where at each step, selected attributes are modeled by naive Bayes and the remainder by the decision table, and all attributes are modelled by the decision table initially. At each step, the algorithm also considers dropping an attribute entirely from the model. For more information see: [36].

FT Functional Trees classifier. Classification trees that could have logistic regression functions at the inner nodes and/or leaves. For more information see: [37].

HyperPipes For each category a HyperPipe is constructed that contains all points of that category (essentially records the attribute bounds observed for each category). Test instances are classified according to the category that "most contains the instance".

Ib1 Nearest-neighbour classifier. Uses normalized Euclidean distance to find the training instance closest to the given test instance, and predicts the same class as this training instance. If multiple instances have the same (smallest) distance to the test instance, the first one found is used.

Ibk K-nearest neighbours classifier. Can select appropriate value of K based on cross-validation. Can also do distance weighting. For more information see: [38].

J48 Pruned C4.5 decision tree. For more information see [39].

Jf8Graft Grafted, pruned C4.5 decision tree. For more information see: [40].

JRip Implementation of a propositional rule learner, Repeated Incremental Pruning to Produce Error Reduction (RIPPER). For more information see: [41].

KStar K* is an instance-based classifier, that is the class of a test instance is based upon the class of those training instances similar to it, as determined by some similarity function. It differs from other instance-based learners in that it uses an entropy-based distance function. For more information see: [42].

LMT Classifier for building 'logistic model trees', which are classification trees with logistic regression functions at the leaves. For more information see: [43].

Logistic Multinominal Logistic Regression model with a ridge estimator, a slightly modified implementation of the algorithm described in [44].

LWL Locally weighted learning. Uses an instance-based algorithm to assign instance weights (a brute force search algorithm for neirest neighbour). Uses DecisionStump as base classifier. For more information see: [45].

MaxEnt WEKA Implementation of the Stanford Maximum Entropy Classifier[1].

MultilayerPerceptron A Classifier that uses backpropagation to classify instances.

NaiveBayes A Bayesian Network classifier that assumes independence of its input features. For more information see: [46].

NaiveBayesSimple Naive Bayesian Network where numeric attributes are modelled by a normal distribution. For more information see: [47].

NaivesBayesUpdateable A Naive Bayes classifier using estimator classes. This is the updateable version of NaiveBayes.

NBTree Decision tree with naive Bayes classifiers at the leaves. For more information see: [48].

NNge Nearest-neighbor-like algorithm using non-nested generalized exemplars (which are hyperrectangles that can be viewed as if-then rules). For more information see: [49].

OneR A 1R classifier; in other words, uses the minimum-error attribute for prediction, discretizing numeric attributes. For more information see: [50].

PART A PART decision list. Uses separate-and-conquer. Builds a partial C4.5 decision tree in each iteration and makes the "best" leaf into a rule. For more information see: [51].

RandomForest A Forest of Random Trees. For more information see: [52].

RandomTree Class for constructing a tree that considers K randomly chosen attributes at each node. Performs no pruning.

RBFNetwork Normalized Gaussian Radial Basis Function Network. Uses K-means clustering algorithm to provide the basis functions and learns a logistic regression on top of that. Symmetric multivariate Gaussians are fit to the data from each cluster. If the class is nominal it uses the given number of clusters per class. It standardizes all numeric attributes to zero mean and unit variance.

REPTree Fast decision tree learner. Builds a decision/regression tree using information gain/variance and prunes it using reduced-error pruning (with backfitting).

[1] http://nlp.stanford.edu/downloads/classifier.shtml

Ridor RIpple-DOwn Rule learner. It generates a default rule first and then the exceptions for the default rule with the least (weighted) error rate. Then it generates the "best" exceptions for each exception and iterates until pure.

SimpleCart Class implementing minimal cost-complexity pruning. For more information see: [53].

SimpleLogistic Classifier for building linear logistic regression models. LogitBoost with simple regression functions as base learners is used for fitting the logistic models. The optimal number of LogitBoost iterations to perform is cross-validated, which leads to automatic attribute selection. For more information see: [43].

SMO-Polykernel Implementation of John Platt's Sequential Minimal Optimization algorithm for training a support vector classifier. Uses a Polynomial Kernel.

SMO-RBFKernel Implementation of John Platt's Sequential Minimal Optimization algorithm for training a support vector classifier. Uses a Radial Basis Function Kernel.

VFI Classification by voting feature intervals. Intervals are constucted around each class for each attribute (basically discretization). Class counts are recorded for each interval on each attribute. Classification is by voting. For more information see: [54].

Votedperceptron Implementation of the voted perceptron algorithm by Freund and Schapire. For more information see: [55].