Predicting dialogue state transitions using prosodic markers

Exploring AMI Corpus backchannels

I.C.C. Nouwens, s9909273

27-08-2009

dr. D.K.J Heylen
Department of Human Media Interaction, University of Twente

dr. ir. H.J.A. Op den Akker
Department of Human Media Interaction, University of Twente

M. ter Maat, MSc
Department of Human Media Interaction, University of Twente

dr. K.P. Truong
Department of Human Media Interaction, University of Twente

Abstract

In a typical conversation held within a small group, we often see one person speaking, whilst the others listen. In most dialogues the listeners are not completely silent while the speaker has the floor. Occasionally they indicate to the speaker and the rest of the group their engagement in the discourse by giving feedback in the form of words like "Oh really?", "yeah", "hmm-mm", "you don't say!". By doing so, the listener informs the speaker with their opinion about what is being said. Opposed to these feedback words – called *backchannels* – a listener can choose to interrupt or take over with a new statement if he or she wishes to contribute more than a few backchanneling words.

In this thesis we have studied recordings of conversations in order to determine if the prosody from the speaker contains characteristic differences between the situation in which a listener uses a backchannel and the other situation, in which a listener adds an entirely new verbal contribution to the conversation.

We used a corpus, consisting of the recorded signals of 138 multiparty meetings with an average length of 33 minutes each, in which four participants discuss the design of a new product. From the participants' speech each utterance is annotated with a type, indicating whether it is a backchannel or not. From this corpus we selected the utterances of a speaker wherein or where shortly after, one of the listeners would start a contribution.

By using "Praat", we extracted several prosodic features from the selected utterances, normalized them for each speaker and used the resulting dataset in a series of machine learning experiments. By applying statistical techniques on our data, we assessed whether the two different types of contributions could be distinguished, based on the prosodic features that can be taken from the speaker's speech. We found our decision tree to be correct in classifying the type of the contribution in backchannels and non-backchannels in 65.9% of all cases. With the baseline set at 50%, this is an increase of 15.9%.

This report will present the following contents: chapter one serves as the introduction and is followed by chapter two, presenting previous findings from fields related to our study. Chapter three will present the corpus used and how this was formatted to support automatic utterance selection. The selection criteria, resulting data selection and feature sets that are extracted are presented in chapter four. The experiments that were conducted and the results obtained from them are described in chapter five. Finally, we conclude in chapter six with conclusions and suggestions of further research.

Foreword

This thesis is part of the requirements of acquiring a Master's degree in computer science at the Department of Human Media Interaction at the University of Twente. It is the final component of a number of years spend as a student at the University of Twente in the Netherlands. Years I already look back upon as being the best years of my life. In this time I have met many new individuals who I now call friends. This does not imply, of course, that I am not eagerly looking forward: after all, things can always get better.

I would like to thank Dirk Heylen and Rieks op den Akker for their insightful guidance over the duration my entire masters program. Besides the thesis they have assisted in several projects and courses throughout the last years. I would also like to thank Mark ter Maat and Khiet Truong, whose comments and insights contributed greatly. In the hectic last few weeks I have often had the privilege of accurate commentary, useful opinions and swift responses.

I would also like to send out a warm heated "thank you" to my family for their general concern and to Arthur Melissen and Joke Noppers for their literary comments and organizational help regarding the festivities surrounding my last day being a student. Last but foremost I would like my girlfriend Joke to know that without her essential support, the final weeks would have been considerably more difficult.

Table of contents

1. Introduction 6

2. Background 7

2.1 Conversation segmentation: the pause feature. 7

2.2 Transition relevant places and prosody 8

2.3 Summary 12

2.4 Definitions and objectives 13

3. Data formatting 16

3.1 The Corpus 16

3.2 A conversation in the AMI corpus 19

3.3 Target identification 21

3.4 Grouping 24

3.5 Experimentation 26

4. Data selection and extraction 28

4.1 The AMI -> Praat parser 28

4.2 Grouping and selecting 31

4.3 Prosodic feature extraction and analysis 37

5 Processing results 44

5.1 Varying window length 44

5.2 Feature sets' results 50

5.3 Increasing the resolution 52

5.4 Filtered slopes 54

5.5 Focus of attention 55

5.6 Evaluation 56

6 Conclusions and further research 59

6.1 Conclusions 59

6.2 Further research 59

References 60

Appendix A1 Dialogue state distribution > 1 prom 63

Appendix A2 DST distribution > 1 prom 64

Appendix B1 Experiment 1.1 65

Appendix B2 Experiment 1.2 66

Appendix B3 Experiment 1.3 67

Appendix B4 Experiment 1.4 68

Appendix B5 Experiment 1.5 69

AppendixC1 Experiment 2.1 70

Appendix C2 Experiment 2.2 71

Appendix C3 Experiment 2.3 72

Appendix C4 Experiment 2.4 73

Appendix C5 Experiment 2.5 74

Appendix D1 Experiment 3.1 75

Appendix D2 Experiment 3.2 75

Appendix D3 Experiment 3.3 77

Appendix D4 Experiment 3.4 78

Appendix D5 Experiment 3.5 79

Appendix D6 Experiment 3.6 80

1. Introduction

For some years now, computing systems are no longer communicated to with a mere keyboard and monitor. In recent years new ways of human-machine interaction have been developed. Take for example speech recognition and synthesis systems that are developed to assist in the everyday use of the personal computer. Presently we see the development of *virtual humans*, supplying the computer with a face to go along with the voice. These virtual humans, or *conversational agents*, as they are also called, are developed to assist in various applications. Giving a virtual tour through a building, assisting the school teacher in learning tasks, representing a system that is interconnected with the hardware in your home. Some examples exist today, some might in the near future.

The examples above have a common vantage point: we (humans) interact verbally with a computer system and we expect this conversation to occur smoothly – preferably as natural as possible – as if the system were another human. Building a conversational agent that quickly and adequately responds to user requests or can interact casually, using everyday topics, obviously is not done overnight. There are major difficulties to overcome before a system can be produced, that knows that it is being addressed, is capable of processing human speech and can promptly form an appropriate response. The difficulties can be grouped into three major aspects: How should the agent interpret it's input?; When should the agent contribute to the conversation?; What should it say?

An integral part of the ones role as a listener, is signaling to the speaker that you (the listener) are (still) engaged in the conversation. In natural dialogue this is called *backchanneling* and these signals can be given verbally as well as non-verbally. A good example of a spoken backchannel is "Hmm-mm", but also think of an occasional nod or turn of the head, a sprouted set of lips, raised eyebrows, etc. Although we might take them for granted, backchannels are of great importance in a conversation. Imagine talking to a virtual human on a computer screen, that uses no backchanneling signals. Is it listening? Is it 'on'?. Or, even worse, the other way around: gazing at you, following you around the room, giving an "Uh-u" every 1½ seconds. Clearly, the optimum lies somewhere in between. Although much progress has been made in the development of natural appearing (embodied) conversational agents, existing systems remain far from being perfect.

We distinguish between situations in which a backchanneling utterance was given and situations in which a participant that was a listener before, started talking. We strive to contribute to the fields by determining if characteristics of the forgoing speech can be used to identify these two situations.

2. Background

In the introduction we showed that this study focuses on distinguishing backchannels from other kinds of utterances in a dialogue. In our approach we will select these moments from a corpus containing a large quantity of conversations and extract a set of prosodic features for analysis. Before we go into how we (de)compose a dialogue with the intent of finding these items and what to extract from them, we elaborate on features and methods used in other studies on closely related topics and how one of the most important attributes of speech came into play: *silence*.

2.1 Conversation segmentation: the pause feature.

In the fields of turn-taking and conversation modeling many researchers refer back to a study from 1974: “Simplest systematic for turn-taking”, (Schegloff, Sacks, & Jefferson, 1974). Although it was a design that would be discussed (and sometimes criticized) in many studies to come, it is also considered to be one of the first models for turn-taking. In the 70’s and 80’s linguists presented many more studies on the properties of human dialogue with the intention of designing better dialogue models, (Jurafsky & Martin, 2000). The notion that sophisticated dialogue systems, used by conversational agents, would need some understanding of human dialogue as well, arose later. In the following decades, the 80’s and 90’s, this lead to many systems that initially relied on keyboard input, producing output on a screen. Later, these also accepted and generated *spoken* language. It wasn’t until the late 90’s and early in the 21st century that extra modalities, like non-verbal communication with hand gestures, or gazing behavior were taken into account in (spoken) dialogue systems.

In the past decade many systems have been developed, that rely mainly on an increasing period of silence at the end of the interlocutor’s turn for the decision on when to start with their answer. This simple systematic can prove very useful for end-of-turn detection in question-answering systems, where there is one person asking the questions, and one agent, attempting to provide the requested information. Classic examples are ticket or flight reservation systems. Preferably the agent’s responding time has the perfect balance between short but possibly wrong, and long but possibly awkward or unnatural. The perfect agent starts as soon as possible, but never when it shouldn’t.

In the first mentioned study, a conversation was modeled as a collection of turns of conversational partners with an uninterrupted sequence of utterances as contents, (Schegloff, Sacks, & Jefferson, 1974). At the end of an utterance the speaker would choose to continue with a new utterance or yield the floor to the partner, who signaled that he or she wishes to contribute. In this manner a dialogue is structured in a very ‘civilized’ manner, in which participants are expected to prevent overlapping turns. It was found that in some occasions the pause between utterances in the same turn were longer than the pauses between utterances of two different turns and noticed a small amount of speech that overlapped with other speech. In these situations, the participants should attempt to repair the dialogue, comparable to a real world scenario in which the speaker is interrupted or two participants start simultaneously. It was mentioned that relying solely on a pause feature would lead to a few errors.

In a different study, it is argued that an interruption or long pause within a turn are not errors in the dialogue but rather part of the normal discourse structure, (O'Connell, Kowal, & Kaltenbacher, 1990). In this view, a conversation is a joint activity between speakers where the dialogue is a result of a common goal of the speakers: both want to convey information or an opinion.

Whether or not an interruption or overlap is labeled as an error in a conversation, they seem to occur quite often. A study about the classification of meeting activities found that a considerable overlap of utterances occurred in the corpus that was used, (Tommassen, 2007). Little under seven percent of all sentences were at some point overlapped by an utterance from another speaker – excluding the backchannels, which tend to occur naturally in overlap with other utterances, considering their intended nature. In cases of interrupts or overlap one cannot rely (solely) on the pause duration as a feature for deciding when to start; there is none.

Apart from the pause related features, there are many more items, that can be extracted, from the speech of conversational partners. These items are in general referred to as *prosodic features* and can vary over tonal or pitch related attributes, duration and intensity of speech elements, or intonation of specific words. These feature's values represent, in principal, a large amount of data that we ourselves unconsciously use in our everyday structuring of conversations, but that is lost in written text. The relevance of prosodic features for the recognition of discourse structures has been found to apply across different ages, sexes, languages and cultures.

In the past decade several studies have presented methods to segment human-human dialogue into smaller items, using prosodic features as a means of identifying the segment borders and possibly classifing the resulting segments into groups as well. Some used prosodic features to identify end of turn boundaries (Jonsdottir, Thorisson, & Nivel, 2008), (Schlangen, 2006); others segmented recordings of continuous speech in natural dialogue into sentences and utterances (Shriberg, Stolcke, Hakkani-Tür, & Tür, 2000), (Kolář, Shriberg, & Liu, 2006), (Atterer, Baumann, & Schlangen, 2008).

2.2 Transition relevant places and prosody

Decomposing speech into smaller segments and labeling them, is necessary for the automatic recognition of these items in general. This decomposing of a turn was already applied by (Schegloff, Sacks, & Jefferson, 1974); they deconstructed a turn into several "turn constructional units" (TCU), comparable to separate sentences. In their turn taking systematic, each ending of a TCU indicates a moment in which the floor can either stay with the previous speaker or switch to another. These moments are called "transition relevant places" (TRP). This term is adopted in the field as the period indicating the end of an utterance or the end of a turn. We shall review a number of studies that focused on the detection (and prediction) of TRP's

In a study directed at the segmentation of speech into sentences and topics, a set of local prosodic features was used to determine if two sequential words surrounded a sentence boundary, (Shriberg, Stolcke, Hakkani-Tür, & Tür, 2000). This, initially large set of prosodic features was paired down to a usable set of well performing attributes. This was done by conducting experiments wherein the functional merit of the feature was determined by decision tree classifiers. Furthermore a set of lexical features was input into language models that used word probabilities around the sentence boundaries. To compare the separate feature sets on the task of sentence segmentation, two corpora containing speech, that differ in content essence, were used. One consists of a body of natural dialogue between two humans (the switchboard corpus, presented by (Godfrey, Holliman, & McDaniel, 1992)). The other contains a corpus of news messages that were read aloud (the Broadcast News corpus, presented by (Graff, 1997)). The performance of the different feature sets were compared as well and it was found that the prosodic feature set performed on par with the lexical set, for both corpora. A combination of both resulted in the best scores. A big advantage of a prosodic feature set is that is can be extracted with relative ease, compared to lexical features, for which a speech recognizer is necessary in a real life scenario.

The resulting models use attributes that measured *around* the actual boundary, and thus one word and a potential pause in between, in the future, when seen from the moment at which the sentence actually ends. Therefore this would be of poor use in a real time scenario. The actual intention was automatic sentence segmentation of a recorded set of utterances, in which scenario the ‘future’ is at the experimenter’s disposal.

In a realtime scenario the near future is understandably ‘off limits’. Many of the current dialogue systems do however rely to some extend on pause and silence related features, because in many cases it is simply found that performance increases when this information is used. In these cases, the decision depends on a preset minimum pause length. (De Kok & Heylen, 2009). It is argued, that by making use of any form of a pause feature that uses any time *behind* the actual boundary (of whichever unit), a transition relevant place recognizing model can never *predict* the actual TRP, only *detect.* After all, it can only use hind sight, so when the threshold of an *x* amount of time is triggered, the actual speech has progressed with a minimum of *x* time. To be able to truly predict a TRP, a model needs to be pause independent.

In a study about classifying spoken words as being the end of a turn or not, (Schlangen, 2006) attempted to bridge the gap between detection and prediction. A mixture of prosodic and syntactic features was used to classify each pause of a certain length as being the end of the turn or not. In a series of experiments this length was shortened until each word had to be classified as being the last of the turn (or not). This enabled measuring of the necessity of the pause feature. The result showed that the use of any pause threshold above 0 did contribute in the f-measure values for the end-of-turn class. Without any use of the pause, the distribution of end-words and not-ending-words became biased towards the latter, making the results difficult to compare. Yet still, they showed a contributing value of the syntactic and acoustic features. The acoustic features were indicated to perform on par with the syntactic features, in correspondence with the findings of (Shriberg, Stolcke, Hakkani-Tür, & Tür, 2000).

A comparable approach was used in another study, determined to break a continuous stream of speech down into utterance segments. The influence of the pause related features was intentionally minimized, (Atterer, Baumann, & Schlangen, 2008). The input consisted of sentences from the switchboard corpus – no longer than 25 words – from which a prosodic and syntactic feature set was extracted. Classification was done word-wise, indicating whether a word was the last of a sentence or not. This time around, different performance results were found: the syntactic feature set had a substantially larger contribution in the performance than the prosodic feature set. This was concluded after separate and combined testing on the classification task.

A very different feature set aimed at end-of-turn prediction was explored in another study, (De Kok & Heylen, 2009). Here, a segment of a natural conversation corpus was used, that has been outfitted with manual annotations of the head movements and focus of attention of the four interacting participants. These features were used next to a prosodic feature set. When just one feature set was used for the classification experiments, the prosodic feature set achieved higher performance scores than the *multimodal* feature set. Classification performance peaked when the combination of both were used, implying an additive value of the head movement and focus of attention feature set.

In the end-of-sentence detecting approach a set of prosodic features was used, that extracted data very local for simplicity and computational complexity reasons. (Shriberg, Stolcke, Hakkani-Tür, & Tür, 2000). The analysis window was determined by two words and consisted of, 200 milliseconds (ms) before the start of the first word and 200 ms after the second. A potential pause between the word was automatically encompassed in the window. Next to this pause feature extractions were done on pitch, voice quality and phone duration. An interesting feature is the pause before the first word, indicating whether the first word continued from continuous speech or if it is the first in a new sequence, because most words in sequence have an intermittent pause of 0 ms. The slope of pitch is styled into a robust contour from which several features depicting general slope, continuity and range. By evaluating experiments using decision trees, the initial number of prosodic features was tuned down to the useful ones. For a corpus consisting of natural dialogue, the important groups were phone and rhyme duration preceding the sentence boundary, pause duration at the sentence boundary and preceding the first word, total duration in the turn and whether there was a speaker turn at the boundary. The useful features that were found in the other corpus (news messages read aloud) are of less importance for this study, given the nature of the contents, but here as well, it turned out that pause and turn duration features are important, as well as the F0 range, gender and if there is a turn boundary at the sentence boundary.

The overall classification performance of these feature sets was compared to the performance of statistical language models, capturing lexical features from the context of sentence boundaries. Speech recognition was used on the spoken words, enabling the use probability models on combinations of boundaries and parts of speech. Because different corpuses were used, a distinction could be made in the merit of different feature sets. It appeared that pitch features contributed more in text that was read aloud and duration and word-based features were of greater use in natural conversation. Pause related features played an important role in both corpora.

In a study into the detection of end-of-turn using pause thresholds, a set of prosodic features was used, that proved useful in a classification task where all words in a corpus containing spontaneous human-human dialogue were classified as being the last word of a turn or not, (Schlangen, 2006). The features that were used, consisted of a series of F0 and intensity measures, taken from 10 ms frames and smoothed over the sample. From these two categories, they took a number of features that depict the curve (direction, number of changes), a set of features normalized to the speaker's overall mean values (min, max, standard deviation), and a set of differences from the mean at boundary frames. The F0 curve was segmented in three sections, where the mean and standard deviation were measured for each of them. Word length was also used. These features were used as a set, showing improvement when included amongst other feature sets, in the classification task. No individual feature performances were reported.

A different approach was found to be useful as well in a study aimed at the detection of detecting end of utterance boundaries. A collection of classifiers was trained and tested with values of a prosodic feature set that consisted of the dot products taken from a set of filters and a set of prosodic extractions of various lengths, (Fuentes, Vera, & Solorio, 2007). A very large set of features was the result: 342 different features per item in the corpus. Although the model was trained on a relatively (considering the amount of features) small corpus of less than a thousand instances, it achieved very nice recall, precision and f-measure scores, with a REP decision tree as best performer. The features included log pitch and energy and were taken over 56 different samples with a length of 50 ms, spanning back to 3 seconds over the utterance. This set could be used efficiently, obtaining good classification scores, by using a set of filters, applied to the separate intervals, creating a smoothed pitch and intensity slope.

In a study about detecting sentence boundaries, a combination of a prosodic and syntactic feature set was used as well, (Atterer, Baumann, & Schlangen, 2008). Comparable to the previous studies, they extracted from prosody: pitch and intensity average and standard deviation values over windows varying between 50 and 5000 milliseconds and differences between them, depicting pitch and intensity slopes. Furthermore the *place* of minima and maxima were included. A syntactic feature set kept track of the number of words and duration of the sentence and consisted of:

- n-gram models depicting the probabilities of trigrams where the last unit is the actual border;
- internal parsers state related features;
- a set that is related to the syntactic parse tree of the sentence.

This latter indicated the part-of-speech (POS) categories for the words and kept track of the amount of nominal phrases and verbs in the sentence, resulting in the probability of a end-of-utterance encounter. It turned out that the n-gram and the POS feature set had a substantially larger contribution to classifier performance than the prosodic feature set.

The merit of a multimodal feature set in the detection of end-of-turn boundaries was also researched, (De Kok & Heylen, 2009). The study used a set of 14 meetings, annotated with the locations to which the participants are looking during conversation as well as the gestures made by the head. These features were compared on performance to a set of prosodic features that specialized in intonation slope features, including slow and fast fall or rise of both pitch and intensity. It was found that the focus of attention and head gestures feature set did not perform as well as the prosodic feature set, but contributed nonetheless.

Other studies have also used prosodic attributes to model backchannel prediction. (Ward & Tsukahara, 2000), For example focused on a region of at least 110 milliseconds at which the speaker produced a low pitch. Furthermore, it seems that, for the detection of backchannels as well, the pause feature has a prominent role. Other work focused on pause durations of no less than 600 milliseconds or more for the detection of backchannels and based their model on these ‘silence’ features and part-of-speech trigrams (Cathcart, Carletta, & Klein, 2003). A decision tree was used for classification. In another approach, a sequential model was made by combining a Hidden Markov Model and a Conditional Random Field that use a varying set of input features. This then produces the probability that a backchannel should occur, given prosodic extractions from a period of speech as input. Next to continuing regions of low pitch they also us continuing downward pitch slopes and periods without speech (pauses), (Morency, De Kok, & Gratch, 2008).

2.3 Summary

In the past decade, many studies have been using prosodic features (among other sets), to detect their respective transition relevant places, whether it be at the end of a specific set of words, a dialogue act, an utterance or a turn. In some studies this proved to be more useful than in others, but in most (if not all) the use of prosodic features had merit in distinguishing between different classes. The features that were found to be of most value, were periods with low pitch values and drops of pitch and intensity, expressed in respective slope features.

With this background, we assume that prosodic, syntactic and multi modal feature sets have merit in finding transition relevant places’ regions. Also we have seen that decision tree based approaches can be used in the classification of TRP’s and to determine the contributing values of the different attributes. Very different input corpora and evaluation methods have been reviewed and it is hard to say which would be best. In this study we will be using a corpus containing speech and corresponding word transcriptions and dialogue segment annotations. It will be presented in the next chapter. From this corpus we will select the relevant utterances based on criteria that identifies utterances that were followed by a listener’s backchannel and utterances that were followed by a new speakers’ contribution from all annotated utterances in the corpus. From this selection we will extract several prosodic features and make their values into a dataset containing both types of contributions. Several different classifiers will be trained and tested on this dataset. Decision tree based classifies have been found to function very fast and with good performance.

Regarding the prosodic feature sets used in previous studies, there seems to be a lot of agreement. In most cases intensity and pitch falls and sharp falls seem to be positively evaluated, as well as a collection of periods with low pitch values. Intonation slopes are mostly smoothed or hammered down into robust shapes. A difficulty herein lies with the amount of smoothing that should be applied. As always, there is the balance between two evils: a slope that is too smooth doesn’t contain distinguishing values, and a ‘raw’ slope needs many descriptors, making it unpractical for the classification task. Durational attributes like length of utterance or length of boundary word are also often used. The pause features remained popular throughout many studies, whether they are used for detection or prediction of boundaries. We also see that words or parts of speech are often counted. Whether they can be considered as syntactical (like n-grams of words) or prosodic (like the duration) features can be debated. In this study we include them into our *prosodic* feature set, as there are no other features that could be considered syntactic and this would be a very poor ‘set’.

A section of the corpus contains annotations of focus of attention, and communicating signals from head movements. The corpus that we will be using is the same as the one using the multimodal features, reported about in the previous section. We will test their merit as well. However, as these signals are annotated in just a small part of the corpus, these will be use in a spate experiment.

We assume an eventual real life scenario classification problem, so no right looking features will be extracted. We have however no reservations on the difficulty that a particular extraction would have in a real life scenario. For example number of words would need a word-boundary detecting algorithm, or a speech recognizer. We use the label of the corresponding dialogue act group, which would require a robust (real time!) dialogue act recognizer, etc. Also some feature set contain a large amount of data for a small sample of time. In a real-life scenario a fast computer would be required in extracting all of them in (very near) real-time.

2.4 Definitions and objectives

In the previous paragraph we have seen methods of finding and identifying Transition Relevant Places, or TRP and that these moments indicate the end of a turn. Figure 2.1 shows a most simple example of a part of a dialogue that contains a transition relevant place. In it we see the turn being taken over by speaker B, from speaker A, when he has finished his utterance. This turn could have been given to B, of simply be taken, simply by starting to talk – either way, there was a turn transition. For an end-of-turn detection problem, the situation becomes more difficult if there had been overlap, i.e. if speaker A continued for a period of time, while B had already started.

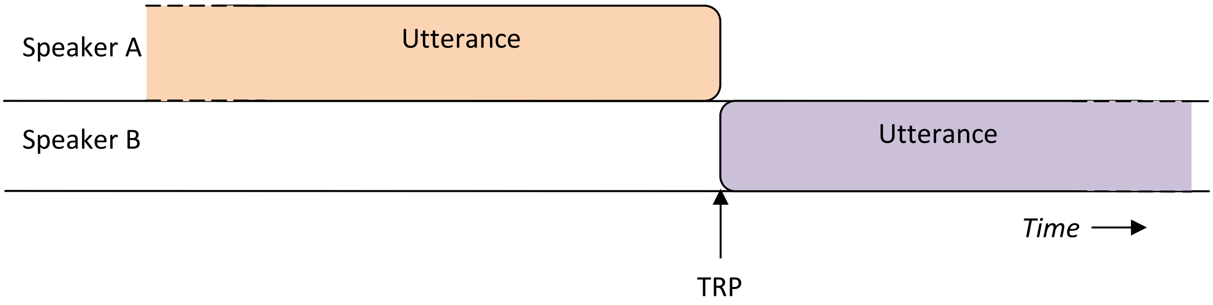


Figure 2.1 A simplest example of a Transition Relevant Place

The difference from the previous example – with or without overlapping speech – and the end of turns in general is that we know of conversational units that *are* utterances following a speaker, but that *do not* take over a turn. In fact, *backchannels* often have the opposite intention. As with end-of-turn detecting tasks we will also be focusing on the moments (and a preceding period of time), where other speakers start with an utterance. But instead of classifying these moments as an end-of-turn or not, we will distinguish between the next utterance being a backchannel or not.

We define all utterances as being *contributions*, so that backchanneling utterances are included and we will define a TRP related term to indicate their starting moment as *Contribution Relevant Places* or CRP.

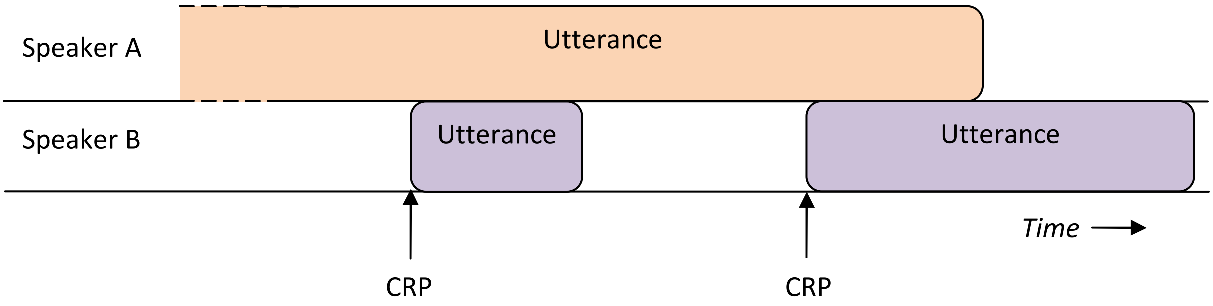


Figure 2.2 Two Contribution Relevant Places

In a new example, based on Figure 2.2 Two Contribution Relevant PlacesFigure 2.2, we see again a speaker A having the floor. In this case speaker B has two utterances, but the floor is kept by A. Without going into detail on the durations of these utterances, this scenario is seen when Speaker B would use a backchannel (B's first utterance) and then attempt to take the floor (B's second, longer utterance). Note that the definition includes the starts of continuing utterances as well. In the previous example, speaker A had one long utterance. If this was subdivided into two or more, the starting moments of those new utterances would have been CRP as well. A new example is given in Figure 2.3. As we will be using the term on more occasions and because we usually refer to just one type, we will subdivide the CRP in two sets and distinguish between them with a type label:

- CRP's[1] that indicate the start of an utterance from a newly starting speaker will be labeled with the type 'n' and referred to as CRPn. The starting speaker thus, is not the one that was already speaking.
- CPR's that indicate the start of a continuing utterance by the same speaker will be labeled with type 'c', so CRPc

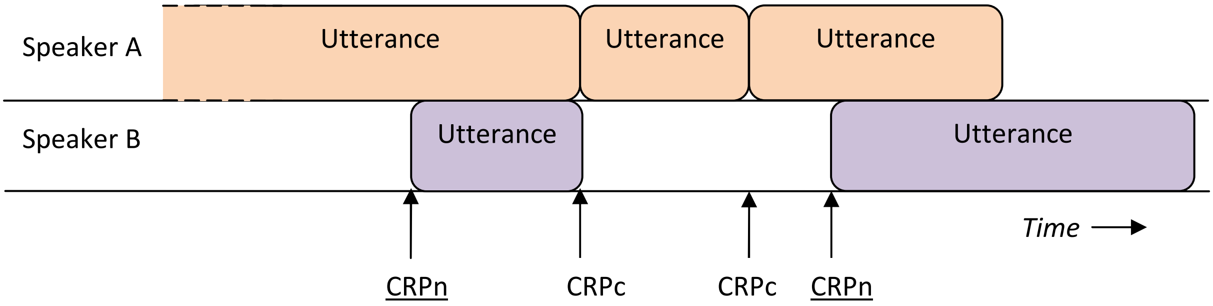


Figure 2.3 Two kinds of CRP: continuing by the same speaker and starting by a new speaker (underlined).

Both types are show in Figure 2.3. We will be using only the CRP that marked the beginning of an utterance by a speaker other than the one that was already uttering – CRPn; these are underlined in the figure.

[1] Generally throughout this report, plural form of abbreviations will be denoted with " 's ". This is done to minimize the mixing with abbreviations that use upper and lower case, or those with an 's'.

We are interested in whether a speaker exhibits certain prosodic characteristics before others in a conversation start to backchannel to him/her and we hypothesize that prosody can give an indication on whether the CRPn following utterance is a backchannel or not. Should this be the case, then these prosodic markers can be included in future conversational models and contribute in finding appropriate moments for conversational agents to use a backchannel. To test this, we will experiment with prosodic feature values that are extracted from the moments preceding a CRPn. For these experiments we formulate the following questions.

Where an utterance that follows on a CRPn is identified as being one of two possible categories: a backchannel or not a backchannel (one of all other utterance types):

- *Can we distinguish between the category of utterances that followed a CRPn, using only prosodic information embedded in the speech preceding the CRPn?*
- *Can a decision model be made that distinguishes between these two categories, given the moment of CRPn and prosody as input and how well would it perform?*

For this experiment we will be using a corpus consisting of multi party natural dialogue, enriched with transcriptions and annotations at word and dialogue act level, containing utterances' boundaries and content. The next chapter will elaborate on this. From this corpus we will select the appropriate contribution relevant places and subject them to prosodic analysis. The resulting extractions form the dataset that serves as input for the eventual machine learning, training and classification task. Next to prosodic features we will also experiment with multimodal features, like the focus of attention and head signals of the speaker and if eye contact was made at the CRPn.

Experiments are conducted, using a subset of utterances, taken from a large amount of data that was annotated beforehand. The resulting subset will still have a considerable size, so the selection needs to be automated. To support this we made a parser that transforms the corpus into a list of items depicting the state of a conversation at any given time. By analyzing the transitions between these states, we can identify any change in the conversation throughout the corpus. This generally usable selection algorithm is applied to find the CRPn and label them as being followed by a backchanneling utterance, or an utterances of another type of dialogue act. Chapter 3 will elaborate on the data formatting and chapter 4 presents the parser.

3. Data formatting

This chapter reports on the corpus that was used in the thesis and how it was formatted to support quick selection of an appropriate subset of utterances. The following sections will explain the setup by answering respectively, the following questions:

[Q1]: What are our requirements, what do we have at our disposal and how is this formatted?

[Q2]: What items will we be using and why?

[Q3]: How can the data best be used and how should it be formatted to support a general setup?

[Q4]: How can we identify and take together similar situations in the data corpus?

[Q5]: How do we proceed towards experimentation?

3.1 The Corpus

As with each study, the quality of the experiments' results rests on the shoulders of the material that was used. This study focuses on prosody in discourse – and, where possible, other characteristics. As chapter 2.5 explains: processing the corpus must be done automatically to be able to generate and use a large dataset. This means that there is need for a corpus, packed with qualitative recording of utterances, that can be parsed easily.

With the goals in mind the choice was made to make use of an existing corpus: The AMI corpus fits the demands (McCowan, et al., 2005), (Carletta, 2007). It is created by the European-funded AMI project whose goal is to improve group interaction by the development of new technologies, based on research on human-human interaction. For studies that are related to the group and their goals, a corpus was created from the recordings from several group meetings, totaling approximately at 100 hours.

The corpus consists of recordings of the proceedings of a number of board meetings. In these meetings, a scenario is played out in which four participants play specific roles of the members of a design team that was ordered to develop a new product: a new TV remote control. Each set of meetings is divided in four separate get-togethers in which the different phases of the design process are elicited: the first shows the project kick-off, followed by functional, conceptional and detailed design. The meetings are held in English, by a mixture of native and non-native speakers of both genders.

The signals that are recorded per meeting, consist of audio and video recordings, capturing all utterances and (most) movements. Audio signals were stored from individual headsets and a recorder in the middle of the table. Video signals consist of recordings taken from cameras at the corners of room and recordings taken from individual cameras depicting the participants face (and upper part of the torso).

The corpus has been enriched with transcriptions and annotations on various characteristics. Spoken words have been manually transcribed and the corresponding dialog acts have been annotated. For clusters of meetings there are also annotations on semantics like topic segments and summaries. Non-verbal cues like focus of attention and movements of the body are included as well for some meetings. The annotations and transcriptions have been encoded in XML format, with good support for parsing the contents. Different items are linked together through unique identities for each type, so that information that is contained in annotations on different levels can be combined. The next sample and Figure 3. 1 show, for example, how a dialogue act is connected to a selection of words, and how all annotations can be connected by temporal alignment.

The following dialogue act encompasses a part of the opening of a random meeting in te corpus:

```
<dact nite:id="ES2008a.A.dialog-act.vkaraisk.3" addressee="D,C,B">
   <nite:pointer role="da-aspect"  href="da-types.xml#id(ami_da_14)"/>
   <nite:child href="ES2008a.A.words.xml#id(ES2008a.A.words3)..id(ES2008a.A.words6)"/>
</dact>
```

The dialogue act is connected to following four words with a matching content of the “nite:id” tag:

```
<w nite:id="ES2008a.A.words3" starttime="32.79" endtime="32.93">Good</w>
<w nite:id="ES2008a.A.words4" starttime="32.93" endtime="33.18">morning</w>
<w nite:id="ES2008a.A.words5" starttime="33.18" endtime="33.86">everybody</w>
<w nite:id="ES2008a.A.words6" starttime="33.86" endtime="33.86" punc="true">.</w>
```

At the same time these focus of attention elements were annotated (like the words and head signals they have their own starting and ending time indices:

```
<foa     endtime="33.56" starttime="32.44" type="person" role="ID" nxt_agent="B"
         nite:id="IDIAP_ES2008a_A.foa.21"/>

<foa     endtime="34.28" starttime="33.56" type="place" place="table"
         nite:id="IDIAP_ES2008a_A.foa.22"/>
```

The annotations of head signals at the corresponding time:

```
<head    endtime="43.798" starttime="0.0" type="no_comm_head"
         nite:id="tmigaz_ES2008a_A.head.1"/>
```

This can be schematically represented, analogues to the figures in chapter 2, by representing each item as a block on a tier (the horizontal bar), progressing in time when read left to right.

The AMI corpus is used in this study as well. Not much of a surprise after listing the upsides above, but it is a reasonable choice: There is a very large amount of data available, recorded in relatively good quality from a wide variation of participants. Also, annotations and transcriptions are encoded in a data structure that can very easily be read: Not only does the AMI project support good parsing functionality, we also have access to a few tools that were created for earlier work.

During the corpus study issued beforehand, a few downsides and limitations on form and quality came to light, that affect the exact choice of the corpus’ usable content, regarding to the data processing tasks at hand. These are listed below, after which the subsection that is used in this study, will be presented.

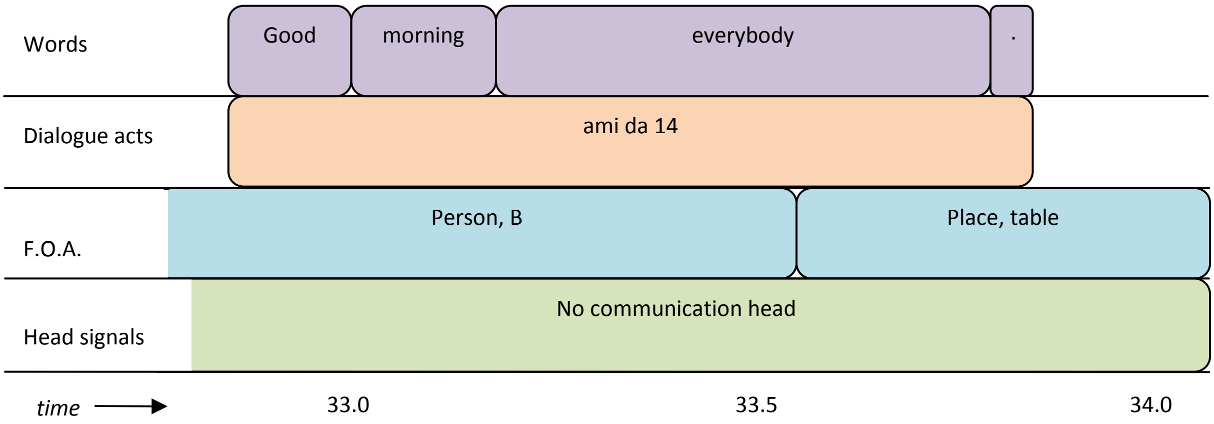


Figure 3. 1 Schematic representation in time of usable AMI corpus annotations

The corpus is enriched with audio recordings from two devices: a recorder located in the middle of the table and a microphone in each participants' headset. The latter accounts for two formats that can be used: a mixture of the utterances from all participants (1 signal) and the separate recordings of each individual's utterances (4 signals). Of these three possibilities, only the separately recorded signal can be used, for two reasons:

- In the mixture of signals, there is a very large difference in loudness between speakers. Although this can be counteracted by normalizing for separate participants, there is an undesired difference in the signal-to-noise (SNR) ratio between loud speakers (higher SNR) and soft speakers (higher SNR).
- On frequent occasions participants are laughing, coughing, or even breathing too close to the microphone. These sounds interfere greatly with utterances that need to be analyzed.

A small drawback from using the separately recorded audio, is that this complicates the analysis process because the identity of the speaker needs to be known, next to already required start and stop times of the audio sample.

The minimum requirements of data that a meeting must contain, before it can be used for the experiment in this thesis are: a qualitative recording of each participants' utterances, an accurate transcription of words, fitted with time indexes and the manual annotation of dialogue acts. The collection of meetings, listed below in Figure 3. 1, conforms to these requirements. This listing is used in all experiments where the prosodic feature set is evaluated. In this table, the total recorded time is the sum of the four meeting recordings, as the signals have been processed separately.

Set of meetings	# meetings	# dialogue acts	# recordings	Rec. size	Rec. time (h:m.s)
ES2002 ~ ES2016	60	47299	240	13.1 GB	131:23.04
IS1000 ~ IS1009	38	26932	152	7.65 GB	80:34.12
TS3003 ~ TS3012	40	42747	160	10.4 GB	97:12.20
Combined	138	116978	552	31.2 GB	309:09.36

Table 3.1 Corpus summarization

Chapter 1 reported about a study that showed a multimodal feature set consisting of annotated head movements and focus of attention, which had a positive effect on classifier performance, (De Kok & Heylen, 2009). These feature sets were also extracted from the AMI corpus. In this study we will also conduct a separate experiment that uses these features. For this experiment, we are limited to a much smaller subset of meetings, since not all meetings have been augmented with annotations on these modalities. In the corpus, there are 13 meetings, that contain the annotations of focus of attention and movements of the participant's head:

- ES2008a,
- IS1000a
- IS1001a, b, c
- IS1003b, d
- IS1006b
- IS1008a, b, c, d
- TS3005a

Set of meetings	# meetings	# dialogue acts	# foa	# head	# recordings	Rec. size	Rec. time (h:m.s)
Combined	13	8426	27724	10670	56	2.4 GB	26:18.06

Table 3.2 Multimodal meeting set summary

The research approach that will be described in the following sections is designed to work with this corpus. The goal of this section was to get the reader acquainted with the AMI corpus and to understand the basics of the data that will be used. Apart from why the AMI corpus was chosen we now also know:

- What data corpus and which subset of meetings will be used to form the base;
- Which annotations from this corpus are used and how they look

3.2 A conversation in the AMI corpus

Chapter 2 explained the general use of dialog acts and the previous section reported that they will be used in our approach as well. Now we shall describe the dialogue acts that are used in the AMI corpus, show how a conversation can be visualized and give a few implications for analysis.

The AMI corpus is enriched with the annotation of dialogue acts for 138 meetings. In these annotations, there are 15 different types of dialog acts, distributed over 6 classes (5 + 1 bucket class). All separate dialogue acts are considered to be a *contribution*, where the backchannel (named specifically in section 1.3) as being a non-trivial contribution. The names, and labels used in the corpus are tabulated later in this chapter, in Table 3.5.

By definition, a conversation is an interchange of thoughts, information, etc, communicated orally. So one can only speak of a conversation when two or more participants are involved. As explained, the corpus depicts four participants in a conversation. We can visualized a conversation in figures

3.1, containing all signal for one speaker, or like figures 2.1 to 2.3 containing speech of two participants. This can be extrapolated to all the four participants, creating an image with 4*4 tiers of blocks (including f.o.a. and head signals), or 2*4 if just words and dialogue acts are used. It is assumed that the reader is familiar with these representations; an example is given in Figure 3.2, showing just two participants. From here on no punctuations are included in the figures (they are annotated as well in the words corpus) as they have no duration, and thus no prosodic contribution (in the corpus). The dialog acts in this figure mark each conversational contribution with a block that has a start, end and type: data that is encoded in each combination of dialog act and first and last corresponding words.

>> A: “Good morning everybody.”

>> B: “Good morning!”

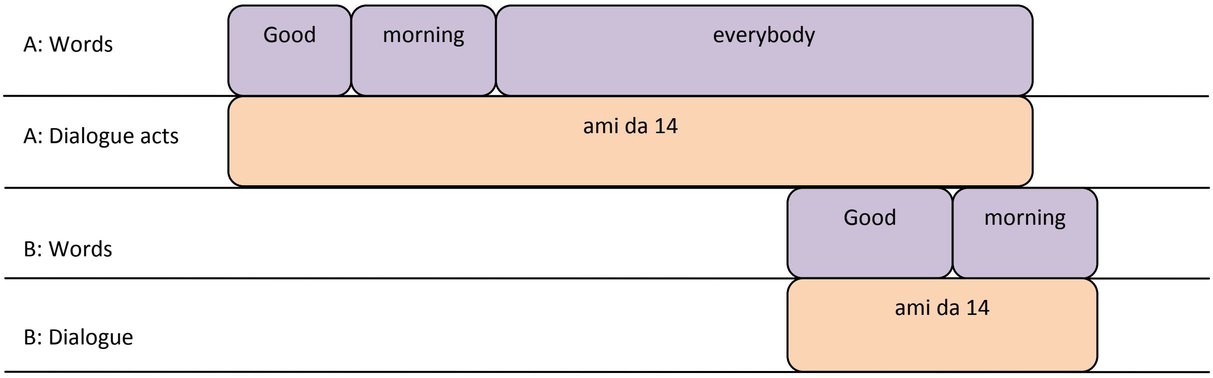


Figure 3.2 A two participant’s dialogue with words and dialogue acts

The thesis sets out to find markers that indicate the type (a backchannel or something else) of a new contribution. This means that the period of speech that a speaker uses right up to the moment at which a new dialog acts starts, is now the area of interest: this is the section in which the markers – should they exist – reside.

Chapter 2 explained that we have an interest in the speech preceding the contribution by the new speaker, marked by the CRPn label. As we are focusing on one particular channel at a time and because separate recordings were used, the speaker that is to be sampled must first be found. There is a small set of meetings for which the dialogue acts do contain addressing information. In these cases annotators have augmented most of the dialogue acts with a target label, identifying one, two or all of the other participants as being addressed by the corresponding utterance. If such a dialogue act is found to follow on a CRPn point, finding the previous speaker would be straightforward. However, since the larger part of the AMI corpus has not been outfitted with this information, finding the correct audio is a bit of a challenge and requires looking at the dialogue as a whole at that moment.

3.3 Target identification

We developed a means of selecting particular moments from the corpus for two reasons:

- All items that will be analyzed – items preceding CRPn – are but a subset of the corpus. These items are subdivided in our two groups: utterance preceding a backchannel and utterances preceding another type of dialogue act.
- For each of these items, we need to determine the previous speaker, since the larger part has no addressing information embedded. Only situations in which the previous speaker can be identified unambiguously, are included.

The selection process is split up into three tasks:

- Produce a tag set that describes each situation in conversation.
- Process each situation into a label from this tag set, that denotes the change in conversation.
- Automate this label production so that relevant situations can be sorted from the rest.

We shall first focus on a tag set. Our primary interest lies in the features that can be extracted from utterances. The boundaries of these utterances are marked by the boundaries of the annotated dialogue acts, that are available in the AMI corpus. Since dialogue acts are directly connected to the words that contain the eventual needed time indices, keeping track of the dialogue acts in the conversation is both necessary and sufficiently to be able to look up any segment of speech from the corpus. Thus, (for now) the outspoken words can be left out from the original dialogue visualization. What remains, are four channels of dialog acts, shown in Figure 3.3.

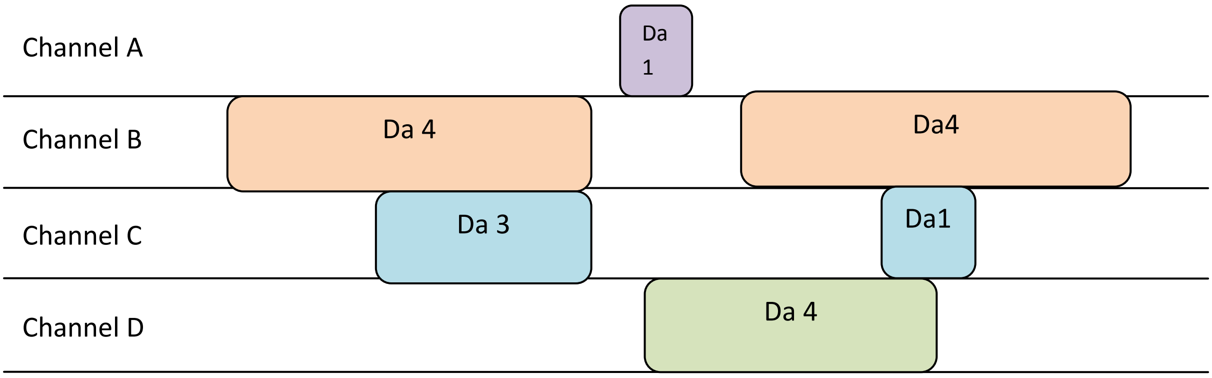


Figure 3.3 DA representing a conversation

The figure shows, that the state in which the dialogue resides, is defined by the combination of all four dialogue acts, at any particular moment. We shall use the term *dialogue state* when referring to any of these situations. This term was adopted from earlier work on multiparty dialog analysis by a study in which an effort was made to identify the different activities a meeting would be in given a sequence of conversational states. (Tommassen, 2007).

We define the status of a conversation by combining all dialogue acts at a given time into the dialogue state:

The dialogue state is a 4-tuple, consisting of all four dialogue act types at one particular time.

To prevent information loss and maintain 'readability' of the state, the label for the dialog state will be made as a four dimensional vector, wherein the dimensions correspond to a participants' dialogue act channel. The value at each dimension corresponds to a dialogue act type.

$(X_i, X_{i+1}, \ldots, X_n)$, where $X \in \{ Da1, Da2, \ldots, Da16 \}$ and i … n correspond to a dialogue act channel

Note that there are 15 different dialogue act types in the AMI corpus, numbered 1 to 16 with number 10 left out. We abbreviate the dialogue act type names to the corresponding type number and in our scenario there are only 4 channels. If a dialogue acts stops, type 0 is appointed. This type does not occur naturally in the AMI corpus and will be used to imply silence on the corresponding channel, so there are 15 + 1 different types of dialogue act and the definition becomes:

(X_A, X_B, X_C, X_D), $X \in \{ 0, 1, \ldots, 16\} \setminus \{10\}$

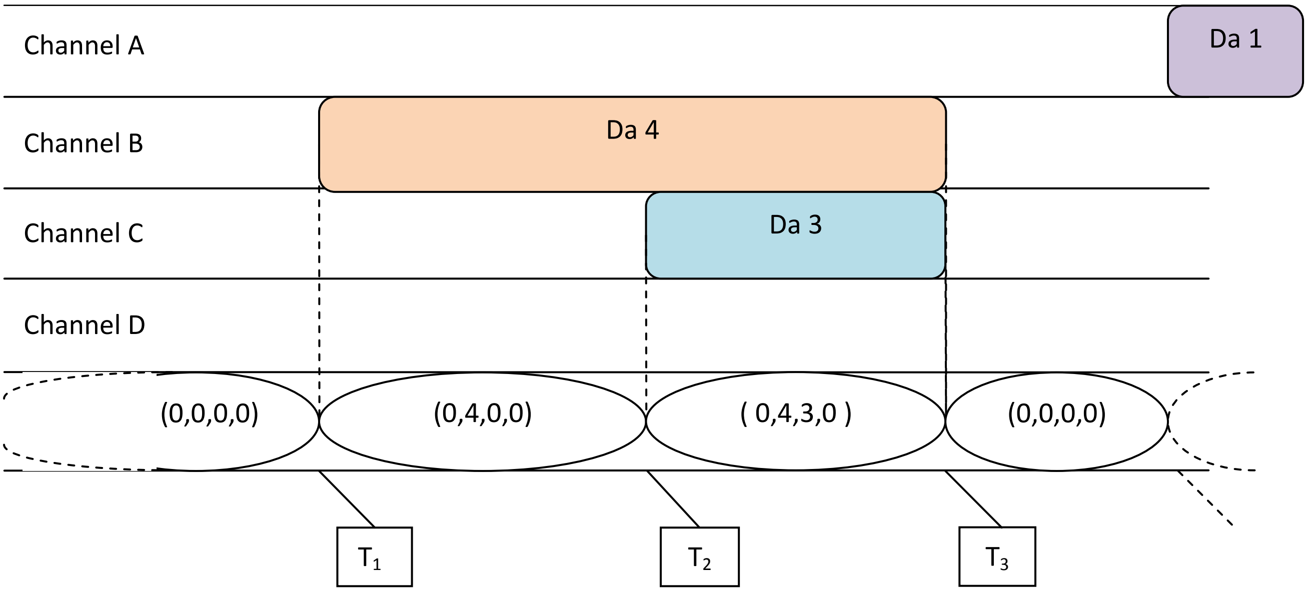


Figure 3.4 Dialogue states and the moments of their transitions: T_i

In Figure 3.4, above, we see an enlargement of a part of the previous dialogue. Without loss of information, the dialogue act channels have been condensed to a single, more complex layer: the *dialogue state*. At this point the changes in a dialogue, can be seen as the transition from one dialogue state into the next. For example: someone starts or stops with an utterance, possibly interrupting another speaker. In Figure 3.4 we see three moments T_1, to T_3, that represent the instance in time at which the dialogue states changes. The term *dialogue state transition* is introduced as a tuple of two dialog states.

DST: $((X_l,X_{l+1},\ldots,X_n), (X_r,X_{r+1},\ldots,X_m))$ where $X \in \{ 1, 2, \ldots, 16 \}$
and l … n and r … m correspond to a dialogue act channel.

$T_1 = ((0,0,0,0), (0,4,0,0))$

$T_2 = ((0,4,0,0), (0,4,3,0))$

$T_3 = ((0,4,3,0), (0,0,0,0))$

In a more dynamic approach, a dialogue state transition can be seen as a transition function on a dialogue state and one or more changes.

Define *action* 'a' as being the *start* 'S' of a new dialogue act, the *stop* 'T' or the *Continuation* 'C' of an already occurring dialogue act, the *switch* 'Wo' to a new dialogue act of the same dialogue act type (a speaker may utter several dialogue acts sequentially) or 'Ws' as the *switch* to a new dialogue act of a different type.

$$a \in \{S, T, C, Wo, Ws\}$$

Define *type* 'da' as the type of the new dialog act.

$$da \in \{\{0,*\}, 1, 2, \ldots, 16\}$$

Define a *channel* c as being one of the speakers identified with A, B, C or D in the meeting uttering the (new) dialogue act:

$$c \in \{A, B, C, D\}$$

Define a *time* index as t in seconds, with an accuracy of 10 milliseconds.

Now Define a *change* g as a four-tuple of these new terms

$$g \in \{(a, da, c, t)\}$$

Finally, define a transition T as a collection of one more changes.

$$T \text{ -> } g^+$$

Now all the dialogue acts in a complete conversation from the AMI corpus can be produced with the starting dialogue state, which is always (0,0,0,0) and a list of transitions:

$$DS(i+1) = DS(i) \text{ x } T(i)$$
$$DS(0) = (0,0,0,0)$$

Note that allowing a collection of changes for each transition allows for the possibility of simultaneous changes in the conversation. (These situations are common.) The example in Figure 3.4also has 2 transitions at T_3. Since it is impossible for a specific speaker to have more than one change, or – in other words – since all changes in a transition will have a unique channel, they have the quality of transitivity and therefore it does not matter in which order they are given, for any transition.

If we look at Figure 3.4 once again, we can exemplify the formalization for the transitions T_1 to T_3. In this case the starting dialog state DS(0) is set to (0,0,0,0) and the moments of T_1 to T_3 are t_1 to t_3 respectively.

$$\mathrm{DS}(1) = \quad (0,0,0,0) \ \mathrm{x} < (\mathrm{S},4,\mathrm{B},t_1) > \quad = (0,4,0,0)$$

$$\mathrm{DS}(2) = \quad \mathrm{DS}(1) \ \mathrm{x} < (\mathrm{S},3,\mathrm{C},t_2), (\mathrm{C},4,\mathrm{B},t_2) > \quad = (0,4,3,0)$$

$$\mathrm{DS}(3) = \quad \mathrm{DS}(2) \ \mathrm{x} < (\mathrm{T},3,\mathrm{C},t_3), (\mathrm{T},4,\mathrm{B},t_3) > \quad = (0,0,0,0)$$

The extensive list of dialogue state transitions can now be used to identify situations in a conversation and also distinguish those that need to be analyzed, from the rest. For example if we would like to find all moments on which a stall type dialogue acts was annotated, we could search the list for transitions containing a change in which the action is a S(tart) and the type is a number 2. Before we get to this point there are some practical problems to overcome.

3.4 Grouping

The previous section showed how the dialogue acts are condensed from multiple speakers into a single data layer, and how a transition can be made from one dialogue state, to the next. Although the list of transitions that can be produced at this point brings us close to discussing which changes in dialogue state will be targeted for analysis, there remains a small problem: the list is rather large. Not very surprising, given the list of all possible changes – even, if the time dimension is ignored: there are 5 possible state changes, 16 different dialogue act types and 4 different channels, per change. Then there and any number between 1 an 4 changes. Now combine these to all possible dialogue states consisting, again, of 16 different dialogue act types, times 4 channels… the number explodes. We define the *situation space* as the collection of all possible results from any dialogue state with any transition. This section will attempt to lessen this space.

The fact that this theoretical space is also rather large in practice, comes to light when the corpus is run through a parser, that builds the dialogue state and lists the different occurrences (this program is presented in chapter 3). The same was done for each combination of two sequential dialogue states – amounting to the complete list of occurrences of dialogue state transitions. Appendices A1 and A2 show the distribution of these types for all cases that occurred one in a thousand times or more in total. The table below shows the amount of dialogue state (transitions) throughout the used corpus. These numbers are summarized from the respective appendices.

Distribution	#theoretical possibilities	Total item count	# different Items	# different items > 1 / 1000
Dialogue state	16^4 (64k)	182.320	2.849	130
Dialogue state transition	(16^4)^2 (4g)	182.182	19.541	170

Table 3.3 Dialogue state (transition) variation over the corpus

The problem lies not with processing this information (a distribution of the entire corpus can be generated in less than a minute on a modern (multi core) computer), but with the eventual classification between states: No classifier can be made, that is accurate enough, to distinguish between all these cases when trained only with prosodic information, extracted from these situations – and probably ever. A relief is that a future classifier actually doesn't need to.

A fairly simple solution lies in the realization that several dialog states and dialogue state transitions actually depict the same situation. For example if speaker (or channel) “A” starts with a dialogue act – say, a backchannel – then this transition reflects the same situation as when speaker B starts with the same type dialogue act. For this situation, it also doesn’t matter who of the remaining three channels was (or is) speaking, when A or B start their backchannel. We have just – in a very rough manner – eliminated 2 dimensions from the equation. The same can be done with the time index: for an representation of a particular situation this can also be left out.

For example, the situation of ‘someone uttering a dialogue act of the inform type’, could previously be shown by one of four possibilities: [4,0,0,0], …, [0,0,0,4] and is now shown by [4]. Likewise, we can show the transition ‘someone backchanneling at an inform utterance’ in several manners:

([4] , [4,1])	(tuple of dialogue states)
[4] x < (S,1) >	(dialogue state and dialogue state transition)
< (C,4) , (S,1) >	(only dialogue state transitions)

Thanks to the *continuation* action, the situation can also be depicted, using only dialogue state transitions (the latter form). Note that we will still need to know who was speaking in the first dialogue state to be able to produce an audio sample that must be analyzed. The same applies for the time index. This information is kept in the original dialogue state, but for the label that represents the situation, this information is no longer relevant and is omitted.

Next to grouping ‘who said something after whom’, we also group different types of dialogue acts together, based on their function in a conversation. This however, does imply some loss of information because the type is abstracted to a general function.

Ami_da_1	BC	“BC”
Ami_da_2	Stall	“S”
Ami_da_3	Fragment	“F”
Ami_da_4	Inform	“ISA”
Ami_da_6	Suggest	
Ami_da_9	Assess	
Ami_da_5	Elicit-Inform	“EL”
Ami_da_8	Elicit-Offer-or-Suggest	
Ami_da_11	Elicit-Asses	
Ami_da_13	El-Comment-About-Understanding	
Ami_da_12	Comment-About-Understanding	“R”
Ami_da_7	Offer	
Ami_da_14	Be-Positive	
Ami_da_15	Be-Negative	
Ami_da_16	Other	

Table 3.4 Dialogue act grouping

In the dialogue state (and transition) representation we shall henceforth use the dialogue act group label ‘*dg*’ instead of the previously used dialog act type label, named ‘*da’*.

$$dg \in \{BC, S, F, ISA, EL, R\}$$

If these changes are incorporated in the corpus parser so that it renames the dialog act types to their new group name, the situation space changes significantly.

Distribution	Total item count	# different Items	# different items > 1 promille
Dialogue state	182320	588	45
Dialogue state transition	182320	1655	111

Table 3.5 Dialogue state (transition) variation after grouping

Chapter 4.1 elaborates on the implementation that can produce the entire list of dialogue states (transitions). So, after grouping all occasions where a dialogue act changes on one of the four channels, according to the parser’s results, there are 1655 different situations in the corpus. Think of these as: “one speaker starts with an ISA-act while another speaker was stalling” or “a speaker falls from a fragment into a stall”. The listing of these items forms the input for the selection process that targets audio samples that fit a specific profile, as the next section will report.

3.5 Experimentation

At this point the corpus is refitted to be used as we please: an experiment can be set up. This section will show how we proceed towards results by giving a short, stepwise notification of the remaining actions that are worked out in the remaining chapters. The selection steps are explained using an exemplary case.

(1) Specify a case that compares group X to Y (or more)

Example case: We wish to select a subset of situations for comparison, where group “C” consist of all situations where someone stalled and then continues and group “I” consists of all situations where someone stalled and is then interrupted by another participant.

(2) For each group, specify a list of criteria that must be met to obtain the desired selection.

Group “C” would like all situations in which there is one active channel, for which there is a change of type ‘switch’ from the ‘stall’ group into any other group but another ‘stall’. Remember, that a switch indicates a change on the same channel, so we know for sure that this particular person continued with another contribution than a stall. Remember also, that a change is a 4-tuple, formatted with action *a*, dialogue act group type *dg*, channel *c* and time *t* by (*a, dg, c, t*), where channel and time may assume any value and can be omitted.

$\mathrm{C}: \quad [\mathrm{S}] \quad \mathrm{x} \quad < (\mathrm{Wo}, \{\, \mathrm{BC}, \mathrm{F}, \mathrm{ISA}, \mathrm{EL}, \mathrm{R} \,\}\,) >$

An alternative would be a continuation of the particular type groups “ISA” or “EL”, selected by:

$\mathrm{C}: \quad [\mathrm{S}] \quad \mathrm{x} \quad < (\mathrm{Wo}, \{\, \mathrm{ISA}, \mathrm{EL} \,\}\,) >$

Likewise, group “I” would like all occurrences of DST where there are two changes: one is a continuation of the type “Stall” and one is the start of any dialogue act group.

I: $[\mathrm{S}] \quad \mathrm{x} \;\; < (\mathrm{C}, \mathrm{S}) \;,\; (\mathrm{S}, \{\, \mathrm{BC}, \mathrm{S}, \mathrm{F}, \mathrm{ISA}, \mathrm{EL}, \mathrm{R} \,\}) >$ or, alternatively:

I: $[\mathrm{S}] \quad \mathrm{x} \;\; < (\mathrm{C}, \mathrm{S}) \;,\; (\mathrm{S}, \{\, \mathrm{ISA}, \mathrm{EL} \,\}) >$

(3) Implement these into the parser and select all occurrences that apply.

The formalization allows for an implementation that can process expressions like the ones given above. In which case a tool that is outfitted with any kind of textual input can be created. Sadly, the available time has not allowed this to be created yet. For now, we settle for a hardcoded version. The argument of grouping that leads to the implementation of a few, relatively simple, rules, still holds: a completely different subset of situations can be selected by changing a few conditions in the programming. Chapter 3 will report on how the parser is outfitted with an selection algorithm that selects all occurrences that are relevant for the experiment that was conducted.

(4) Extract from corresponding audio samples all relevant prosodic features.

In this study the phonetics toolkit “Praat” (Boersema & Weenink, 2001) was used to extract prosodic information. Chapter 3 will report on how this was done. The chapter also shows how the list of selections needs to be formatted for easy processing in the Praat prosodic analysis script and how, in its turn, the script formats the extracted values into the input for the next step. Steps (1) to (4) are incorporated in chapter 4 for this study’s experiments.

(5) Attempt to classify between groups based on extractions and assess feature values.

When a dataset is provided from all the audio samples, we use the machine learning toolkit “Weka” (Witten & Frank, 2005), to try to differentiate between the audio samples, using the data and knowledge of which group (“C” or “I”) they belong to. Weka can be used to explore the predicting capabilities of the (prosodic) features.

(6) Incorporate valuable features into (part of) a decision model

If usable results are produces from Weka’s training and testing algorithms, the decision model that was created in the classification process, can be (partly) used to form better models. In our situation (remember the continue/interrupt example) this would result in set of features that could help an agent that can detect stalls from other speakers decide if it is appropriate to start its own sentence or let the speaker continue. In the end, the decision would be based on a number of examples, taken from the AMI corpus. Chapter 5 will discuss classification and results.

4. Data selection and extraction

The previous chapter has described the general approach on how to bring forth an experiment on prosodic information, drawn from the utterances: First of all, by explaining how the corpus can be transformed into a set of situations that need analyzing and secondly, showing how this set is analyzed and processed into a usable model. This chapter first reports about the developed parser, software tools and analysis script. Essentially, this section can be read apart from the research. It shows how the data was processed and how intermittent data was formatted. Then, in the following paragraphs, we will elaborate on the contents of the dataset by reporting the situations that were extracted and the implementation of the different feature sets.

4.1 The AMI -> Praat parser

Each meeting in the corpus has a number of types and a number of participants that are interacting. The corresponding data is stored per type, per participant, so there is an intrinsic connection between a number of files: four for each type that is processed. Except for the content that is being discussed in the meetings, in which remarks are made about 'the previous meeting', there is no reference at all from one meeting to the next. So at our level of analysis (that doesn't include semantics), using exactly one meeting for the maximum scope of a data structure for, seems a logical choice. With this data structuring, parsing the contents of a meeting can thus be done detached from other meetings, creating an 'embarrassingly parallel' workload. The figure below is a part of the class diagram belonging to the parsing tool, showing the connections between the data structures.

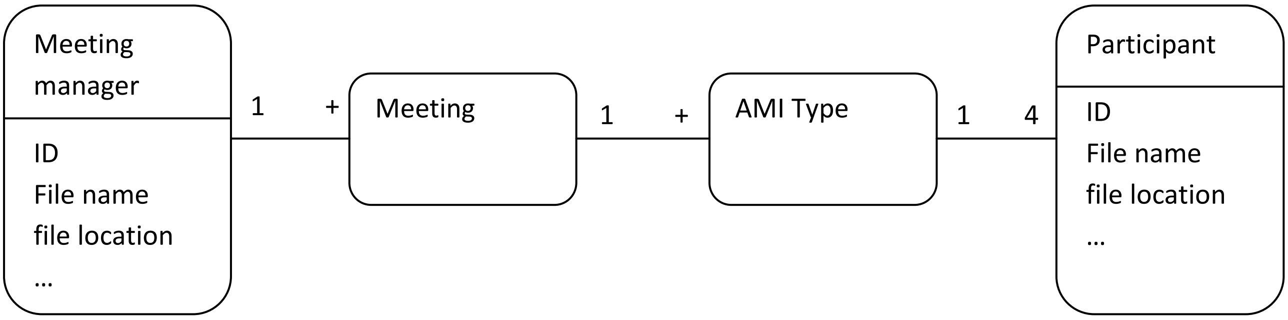
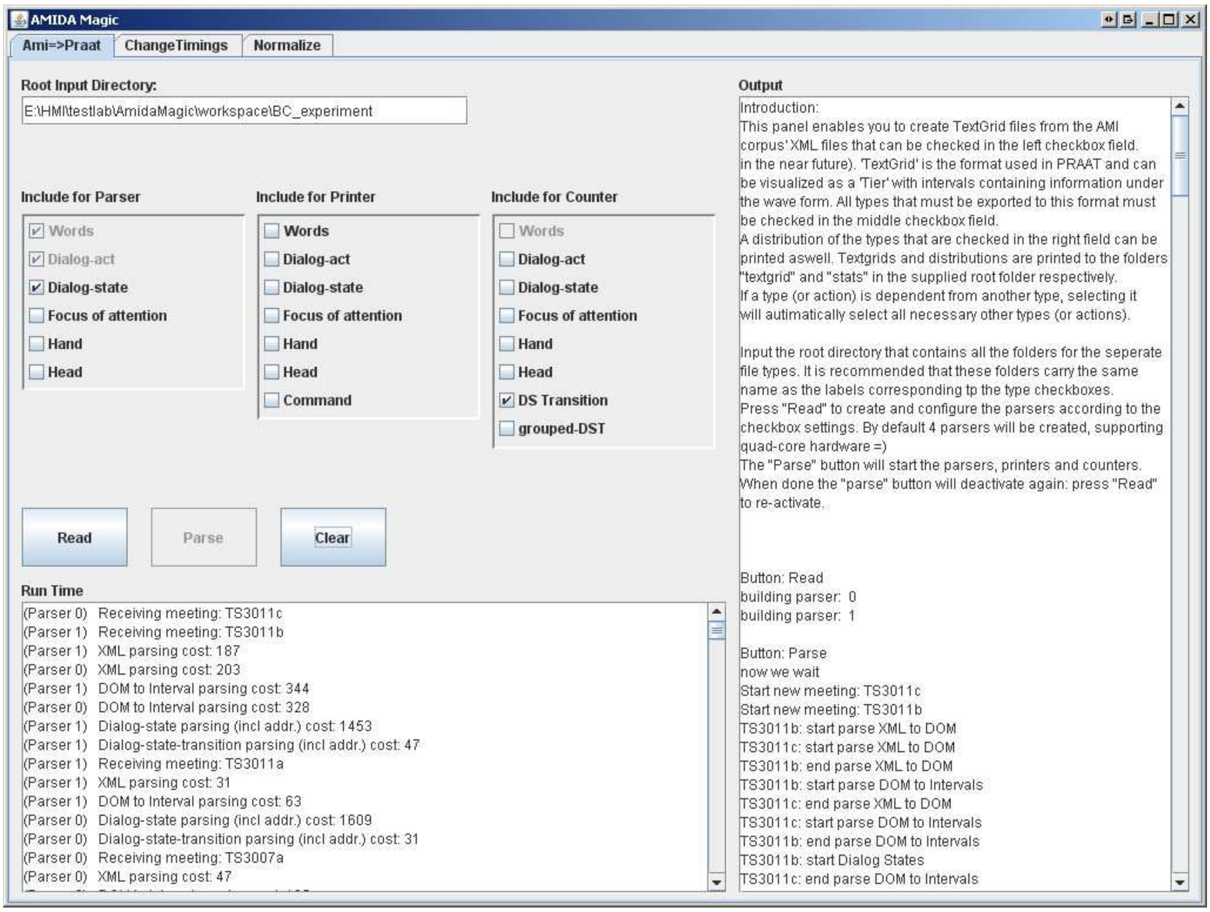


Figure 4.1 Class diagram representation of the connection of corpus contents

into a collection of tools that can be used to parse and change input from the AMI corpus and datasets in other stages of analysis. It has implemented the data structure described above and uses the "AMI -> Praat" parser to create the dialogue state transitions. The graphical user interface for this tool is shown in Figure 4.2.

The front end uses checkboxes to configure the conditions for the parsing process, buttons to issue the actions and text fields to supply the user with information on what was done, how much time is used for intermittent processes and, if error's should occur, in what files and for which process they happened.



Root Input Directory:
E:\HMI\testlab\AmidaMagic\workspace\BC_experiment

Include for Parser
☑ Words
☑ Dialog-act
☑ Dialog-state
□ Focus of attention
□ Hand
□ Head

Include for Printer
□ Words
□ Dialog-act
□ Dialog-state
□ Focus of attention
□ Hand
□ Head
□ Command

Include for Counter
□ Words
□ Dialog-act
□ Dialog-state
□ Focus of attention
□ Hand
□ Head
☑ DS Transition
□ grouped-DST

Run Time

```
(Parser 0)  Receiving meeting: TS3011c
(Parser 1)  Receiving meeting: TS3011b
(Parser 1)  XML parsing cost: 187
(Parser 0)  XML parsing cost: 203
(Parser 1)  DOM to Interval parsing cost: 344
(Parser 0)  DOM to Interval parsing cost: 328
(Parser 1)  Dialog-state parsing (incl addr.) cost: 1453
(Parser 1)  Dialog-state-transition parsing (incl addr.) cost: 47
(Parser 1)  Receiving meeting: TS3011a
(Parser 1)  XML parsing cost: 31
(Parser 1)  DOM to Interval parsing cost: 63
(Parser 0)  Dialog-state parsing (incl addr.) cost: 1609
(Parser 0)  Dialog-state-transition parsing (incl addr.) cost: 31
(Parser 0)  Receiving meeting: TS3007a
(Parser 0)  XML parsing cost: 47
```

Output

```
Introduction:
This panel enables you to create TextGrid files from the AMI
corpus' XML files that can be checked in the left checkbox field.
in the near future). 'TextGrid' is the format used in PRAAT and can
be visualized as a 'Tier' with intervals containing information under
the wave form. All types that must be exported to this format must
be checked in the middle checkbox field.
A distribution of the types that are checked in the right field can be
printed aswell. Textgrids and distributions are printed to the folders
"textgrid" and "stats" in the supplied root folder respectively.
If a type (or action) is dependent from another type, selecting it
will autimatically select all necessary other types (or actions).

Input the root directory that contains all the folders for the seperate
file types. It is recommended that these folders carry the same
name as the labels corresponding tp the type checkboxes.
Press "Read" to create and configure the parsers according to the
checkbox settings. By default 4 parsers will be created, supporting
quad-core hardware =)
The "Parse" button will start the parsers, printers and counters.
When done the "parse" button will deactivate again: press "Read"
to re-activate.



Button: Read
building parser:  0
building parser:  1

Button: Parse
now we wait
Start new meeting: TS3011c
Start new meeting: TS3011b
TS3011b: start parse XML to DOM
TS3011c: start parse XML to DOM
TS3011b: end parse XML to DOM
TS3011b: start parse DOM to Intervals
TS3011c: end parse XML to DOM
TS3011c: start parse DOM to Intervals
TS3011b: end parse DOM to Intervals
TS3011b: start Dialog States
TS3011c: end parse DOM to Intervals
```

Figure 4.2 GUI of the AMI -> Praat parser

The location for the input can be given in the top left input field, this is the root of the input corpus. The tool requires that for each input type, there is a subdirectory with the same name, containing the files of corresponding meetings in this root. For example, if words and dialogue acts are issued for the parser, then are at least these two subdirectories:

\root\words
\root\dialogue-act

The files in these directories are matched to each other by a file manager in to a meeting object, for each meeting. This manager searches for the meeting names that have the format that is used in the AMI corpus so it is required that the files of different types, that correspond to one meeting, use the default meetings names. It is recommended to use the default filenames from the AMI corpus but they may be changed as long as they contain:

- A type name, equal to the directory name and equal to the checkbox name, that orders the parser to include the type.
- the default AMI corpus meeting name.

The file manager kicks into action when the read button is pressed: directories are scanned and pointers to corresponding files are stored per meeting. Should there be missing files for the types that were assigned for parsing, this will result in messages in the output areas for every meeting, that holds inconsistent input.

Furthermore a number of parsers is created, supporting multithreaded parsing. In the current build, this number is hardcoded at 4, since more and more desktop systems are outfitted with quad core cpu's. Creating more parsers than the computer has different cores to its avail, will not speed up the parsing process any further. In fact, since the parsers use a lot of memory (up to 90MB), using more than 4 parsers could slow down the process due to memory swapping and is therefore discouraged. Upon creation, each parser is configured to the wishes of the user according to the (un)checked boxes in the interface.

Once the parse button is pressed, a thread is created for each parser and the start command is given. Active parsers poll the common meeting manager independently from one other for work. The manager holds a queue of meeting objects – which basically are work units for the parsers. When asked for the next unit, the manager takes the next meeting out of the queue. Synchronization of this process prevents situations where different parsers get to work on the same meeting. This process continues until all work units have been handed out and the queue is empty. The architecture of the classic thread pool was used for this design.

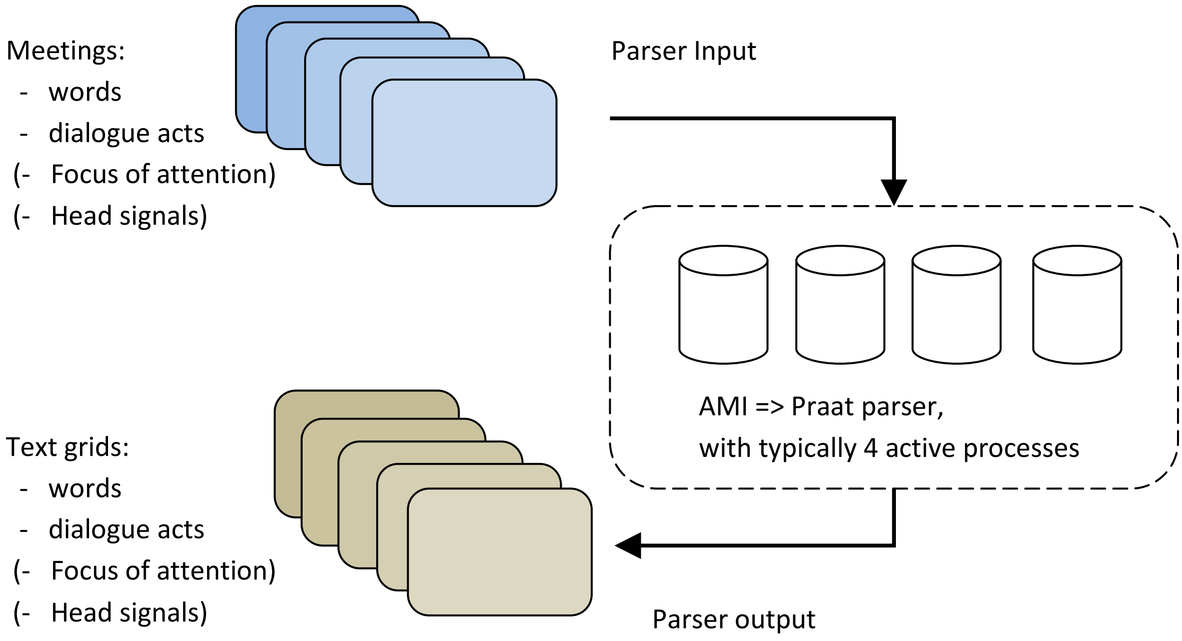


Figure 4. 3 The parser processes the meetings by using a thread pool design

When a parser get's a meeting object – which basically is a structure containing pointers to all the available files for this particular meeting – it reads the relevant XML documents from the AMI corpus and converts the data to maps of intervals. These interval types are an internal data structure that hold (at least) an identifier, start and ending times and a label of the instance they correspond to. This can be a word, dialogue act, dialogue state, or anything that we need to work with. Once the files are completely read, the dialogue acts are used to create interval objects for the dialogue states and dialogue state transitions.

Chapter 3.3 reported about a large number of possible dialogue states and transitions. A distribution of their occurrence can be made when the text for a dialogue state interval is created. This text is given to a 'statsManager' that is commonly used by all parsers. Like the workload distributing this manager accepts the input in a synchronized manner. It sorts the input into the appropriate collection of types (for example the dialogue states) and count the occurrences that have the same text. When all parsers have terminated the distributions, whichever their type, can be printed. This functionality was included initially to support the corpus study and secondly, testing the distributions of groups in datasets. The result of 'measuring up the dialogue states' was already shown in [table … in chapter 2]. To use this option, the user must check the boxes in the "Include for Counter" section in the interface. Each type will produce a text file with a corresponding name in the "\root\stats" directory, containing a distribution of all items that occurred for at least 1 / 1000 times of all items.

In a similar but more simple fashion – synchronization and counting are not required – a list of dialogue states and (more importantly) their transitions can be produced. In its most simple and 'unchanged' form the entries from this list look like these 3 examples:

16.73, 19.56, 19.67, ([4,*,*,*] , [5,*,*,*])

19.56, 19.67, 20.03, ([5,*,*,*] , [5,*,*,9])

19.67, 20.03, 23.32, ([5,*,*,9] , [5,*,*,*])

[preferably: use again from same example (can be manually produced, concept is proven)]

Where the each line is preceded by an identity made up from the meeting and unique dialogue state transition id; secondly the 3 numbers represent starting time of first dialogue state, time of transition and end time of second dialogue state, respectively; the last item represents the transfer. The id and times are necessary for the next application: the extraction process needs to know from which file and at which time indices a sample needs to be extracted.

4.2 Grouping and selecting

We have described the approach, data formatting and parser implementation. In this paragraph, we present the criteria to which the data must correspond and describe the selection that was made from the AMI corpus.

The goal of the eventual experiment is to find prosodic markers, that characterize the type of the following contribution as being a backchanneling type or not. First we will need to distinguish between utterances that were followed by a new speakers' contribution, or those that were followed by a new utterance of the same speaker. In chapter 2 we defined the moment of transition between the previous and next contributions as either CRPc or CRPn. The figure that graphically distinguishes both groups is reused below: Figure 4.4.

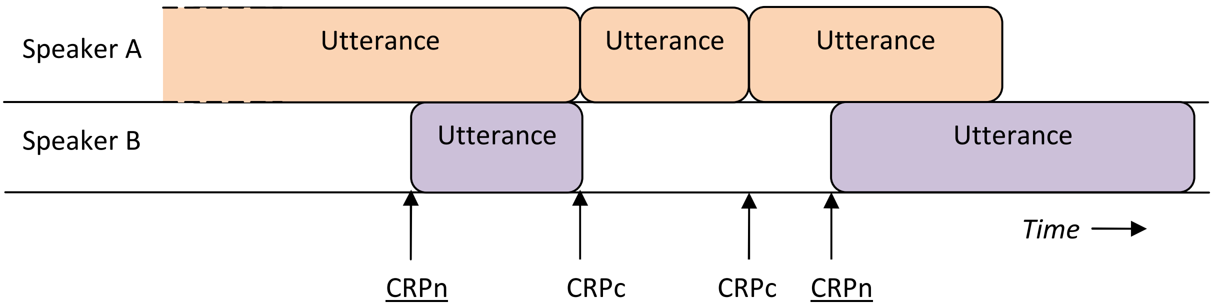


Figure 4.4 CRP's of type *n* and *c*

For each transition that represents an item from the CRPn group, we need to make clear:

- The speaker that is to be sampled.
- The group it belongs to: is the utterance followed by a backchanneling contribution or by another?
- A starting and stopping time of the period of time from which the prosodic features are to be extracted.

If one of these task cannot be performed, the item is discarded. We start by showing the contents of both groups. The set of utterances that was followed by a backchannel is named group "Ba"; the other "NBa". Until now naming and abbreviating terms has (hopefully) not led to any confusion. From this point on, more abbreviations will be used, so to prevent mix-up's we give a short explanation on the labels used in Table 4.1.

Term / abbreviation	Description
"Ba" and "NBa"	labels for the two data *groups* in the experiment, also referred to as *classes*.
"BC"	Used to indicate dialog acts of the type *backchannel* or items from the *backchannel* dialog act group (which, basically is the same).
Participant or speaker A, B, C, D	The meeting participant or speaker, capitals A, B, C or D are used to indicate which one.
"DA"	Abbreviation used for the term dialogue act. Not to be confused with participants A and D
"DS"	Abbreviation used for the term dialogue state.
"DST"	Abbreviation used for the term dialogue state transition.
's	Used whenever a plural form is needed. For example we will be assessing for several DST's if they indicate that the preceding utterance should be labeled as a "Ba", "NBa" or discarded.

Table 4.1 Terms and abbreviations

There are a few limitations on the groups: We are not looking for just all utterances where there the transition contains the start of a backchannel, but for those of which we can be sure that a backchannel was the result. Chapter 2 explained that the audio signals that were used are the separately recorded collection of signals. This means, that there are, three channels that could be sampled, next to the backchanneling speaker. Since, in the largest part of the corpus, there is no addressing information available, we need to fine-tune the selection. Imagine, for example, that there are more than one active speakers and a third (or fourth) participant utters a backchannel. Then, without any addressing information encoded in the BC dialogue act, we can't tell at which dialogue act the BC is directed. In this situation we cannot identify and thus select, the triggering utterance. Some selection tricks are called for: to keep it simple this is done in steps. A nice advantage is that the criterion does not become much more complex by adding a new situation to the selection. We will also elaborate using graphical representations of the situations.

Selecting Ba

For the Ba group all dialogue state transitions are selected that have at least one *starting* DA of the *BC* group.

$$\text{Ba:} \qquad < (\text{S}, \text{BC}) >$$

As told before, it is necessary to have a *continuing* dialogue act, to be able to select the speaker that must deliver an audio sample. This continuing act may be of any type, except for a back channel. Instead of making a large list of possible transitions, a list is used to indicate an option of one of the items in the list. A new change is added to the requirement. This situation is represented in Figure 4.5, below. For simplicity all dialogue acts that are not a backchannel (BC) will be labeled a being "ISA"'s

$$\text{Ba:} \qquad < (\text{S}, \text{BC}) \ , \ (\text{C}, \ \{\text{S}, \text{F}, \text{ISA}, \text{EL}, \text{R}\}) >$$

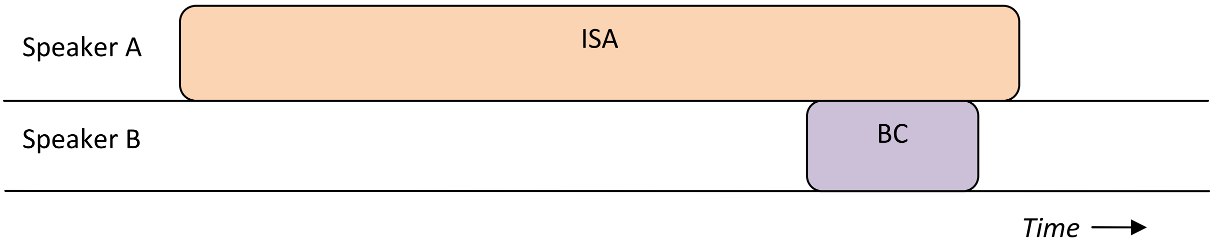


Figure 4.5 The simple backchannel

To increase readability, the collection of all dialogue act groups will be denoted with '(G)' so that {BC, S, F, ISA, EL, R} may be substituted for (G). For the same reason the '-' sign will be used to indicate exclusion, making "(G)-BC" a substitute for all dialogue act groups, except for a backchannel:

{BC, S, F, ISA, EL, R} <-> (G) and {S, F, ISA, EL, R} <-> (G)-BC

We now get:

$$\text{Ba:} \quad < (\text{S, BC}) \;,\; (\text{C, (G)-BC}) >$$

The continuing dialog act may as well have stopped at the instance of the starting backchannel; this is rare, but the speaker can be identified all the same. The 'continuing' type can be expanded with the 'stop' type. A situation that was found to occur often is where the speaker finished, and just after this moment a backchannel was given. With an upper time limit of 250 milliseconds of intermittent pause, these situations are included as well. The pause that can occur between the contributions, this will used as a feature. The end time of the corresponding audio sample is kept at the starting moment of the new contribution. This is represented by Figure 4.6

$$\text{Ba:} \quad < (\text{S, BC}) \;,\; (\{\text{C, T}\},\; \text{(G)-BC}) >$$

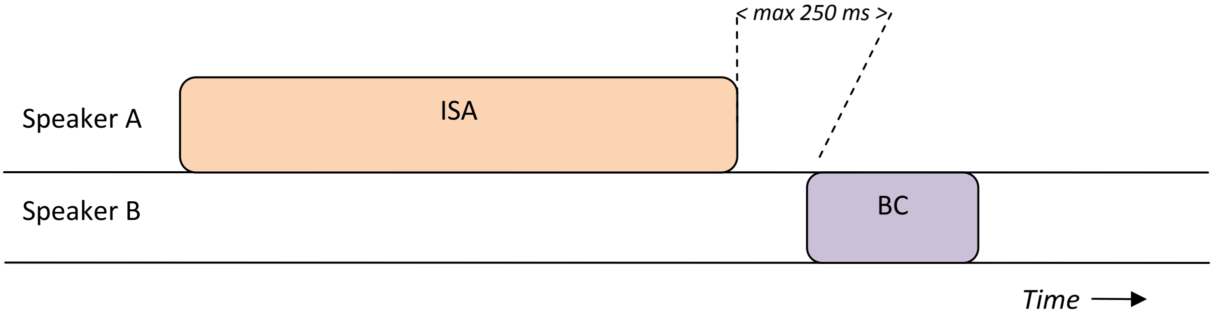


Figure 4.6 Short silences are allowed as well

On that same matter, the starting backchannel may also be a switch from another backchannel: this can happen, if the backchanneling participant utters another one. Note that it must be a switch of the same type (Ws) and not a switch to another type (Wo), because that would result in the same situation as when more than one stopping DA occur in which scenario, we cannot identify the target. This is also represented by Figure 4.7.

$$\text{Ba:} \quad < (\{\text{S, Ws}\}, \text{BC}) \;,\; (\{\text{C, T}\},\; \text{(G)-BC}) >$$

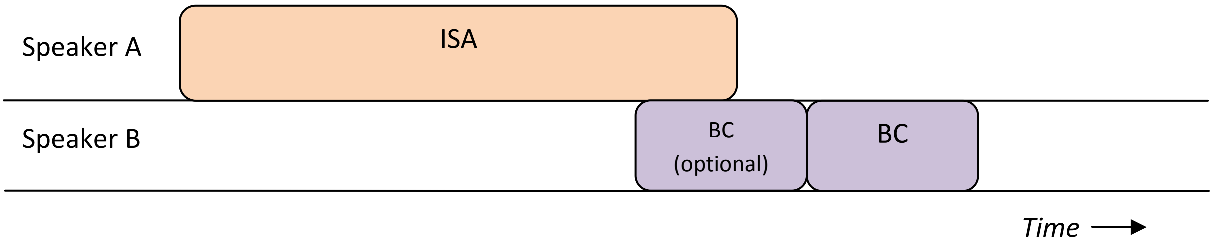


Figure 4.7 With previous backchannels the relevant speaker can still be found

Finally, there may be more backchannels happening at the same moment, since backchannels will not be selected for delivering the audio sample, there could be as many as three, and they may have started already, but could also start at the same moment, or stop, or switch to yet another backchannel. Now this seems to get out of hand, but here as well the possibilities can be added step by step. To indicate multiplicities, a '*' coupled to a change denotes that any number of these changes may be given. In a similar fashion a '+' indicates that a minimum of one of the corresponding changes is required, and more than one are allowed. So

$$\text{Ba:} \quad < (\{S, Ws\}, BC)^{+}, \ (\{C, T\}, (G)\text{-}BC) >$$

indicates that there can be one, two, or thee starting BC's (next to the other requirement of course). To include stopping and continuation of BC, we specify:

$$\text{Ba:} \quad < (\{S, T, C, Ws\}, BC)^{+}, \ (\{C, T\}, \ (G)\text{-}BC) >$$

This seems to be the complete criterion for the Ba collection. But, unfortunately, this criterion now allows for too much. The initial requirement stated that at least one Backchannel started and this is no longer necessary in the current criterion. For example, there could be three stopping BC next to a continuing ISA. This is easily remedied by explicitly naming one stopping or continuing backchannel. We have at least one backchannel that can be given by any speaker. In Figure 4.8 all but one are indicated to be optional. Of course this mandatory one can have any place in this scenario, The backchannels by the other speakers are optional:

$$\text{Ba:} \quad < (\{S, Ws\}, BC) \ , \ (\{C, T\}, \ (G)\text{-}BC) \ , \ (\{S, T, C, Ws\}, BC)^{*} >$$

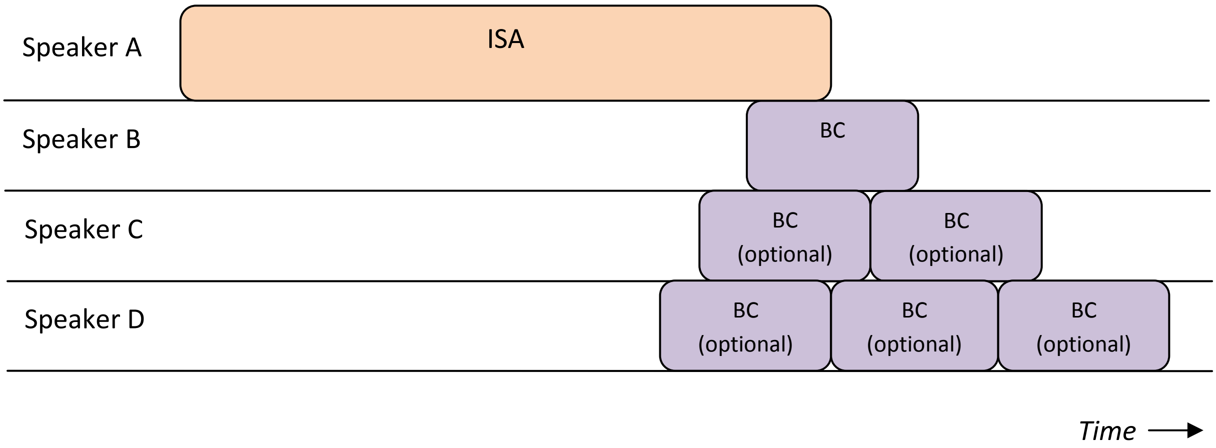


Figure 4.8 A peculiar conversation…

Selecting NBa

The requirement for the other collection, the NBa group, can be constructed in a similar fashion. We start with the minimum requirement: exactly one starting DA that is not a backchannel

$$\text{NBa:} \quad < (S, (G)\text{-}BC) \ >$$

As with the other group, there has to be a speaker that is currently speaking, or has just finished speaking. The dialogue act group does not matter. As with the Ba group, there may be a silence with a maximum duration of 250 milliseconds in front of the new contribution. The situation is represented in figures. Note that the new contribution can be of any type but a backchannel. To denote this we use the same group denoting label as in the critera.

$$\text{NBa:}\quad < (\text{S}, (\text{G})\text{-BC})\ ,\ (\{\text{C}, \text{T}\},\ (\text{G})\ )\ >$$

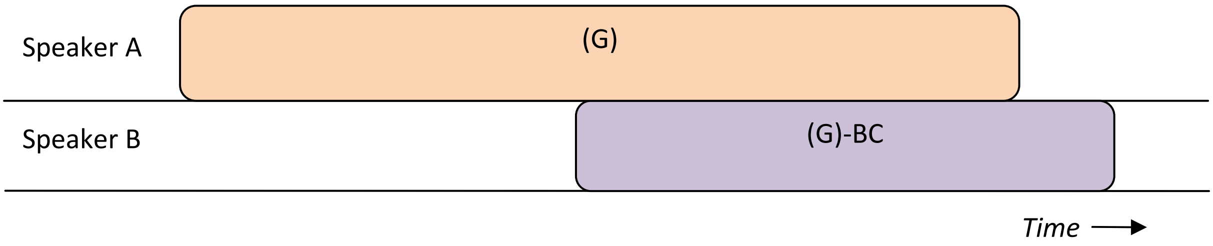


Figure 4.9 The default NBa situation

This already completes the criterion for which the transitions must apply. Relatively simple, compared to the previous group, due to the lack of addressing information. After all, if any more (speaker) channels contain non-BC dialogue acts, we can no longer unambiguously appoint the speaker that must be prosodically sampled.

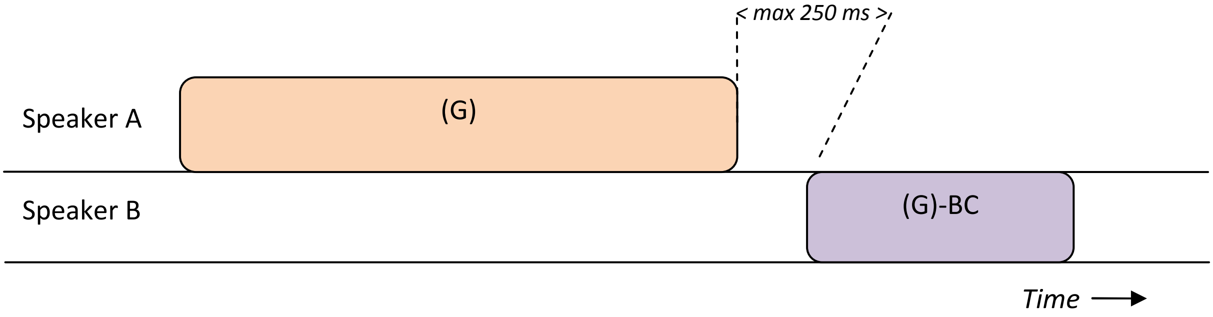


Figure 4.10 Short pauses are allowed as well

The null group

All occurrences that do not comply to one of both requirements are collected in group “null”. This group mainly consists of transitions that indicate a stopping dialogue act, transitions that are not included in the CRPn definition and those for which no speaker could be appointed. The selection process was their first and last attempt for glory: they will not be input into the prosodic analysis script, saving a considerable amount of time

When these requirements are issued in the parser and applied on the entire list of dialogue state transitions, the following numbers are produced:

Class	# Occurrences
Ba	9110
NBa	33867
null	139343

Table 4.2 Post selection class ditribution

The major part of the experiments were conducted on a balanced distribution of both groups. In the parsing process, the selection algorithm is augmented with the option of creating a balanced set, so that any distribution can be made. The algorithm functions as following: The parser processes the DST's and keeps track of the Ba and NBa collection's size. It has been told that the Ba group will be the smallest, which means, that every item that is labeled as a Ba-member, must be accepted. Every time an NBa is processed, a choice is made to accept or discard it. This choice is based on a random number from null to the Ba size at that moment, which is compared to the size of the NBa group at that point and a weight. In this way, the groups grow evenly throughout the parsing process and the NBa items are randomly chosen from the entire group – differently each time. The eventual distribution depends on the value of the weight (default is 1.00). If this set is balanced the number of NBa items will be approximately the same as the number of Ba items – around 9.1k.

4.3 Prosodic feature extraction and analysis

This paragraph reports on the features that were extracted by the Praat script. The attributes, as they are also called, are distributed over different sets, containing different types. The choice to group them together was made to allow for a reasonable number of experiments; they can be compared set wise. They shall be presented in this section in the same manner. We start by defining the terminology that will be used.

Section 3.3 reported that Praat extracts feature values from samples, taken from a larger source of audio. These samples can have any length, as long as they remain within the limits of the audio file itself. The different feature sets that will be extracted for each dialogue state transition, will be varying in length. Each period that is used to denote the start and end time of a sample, is called an interval. In the feature lists, time indices will be used to denote their starting and ending time. With the help of the Figure 4.11 and Table 4.2, we shall explain the used terminology.

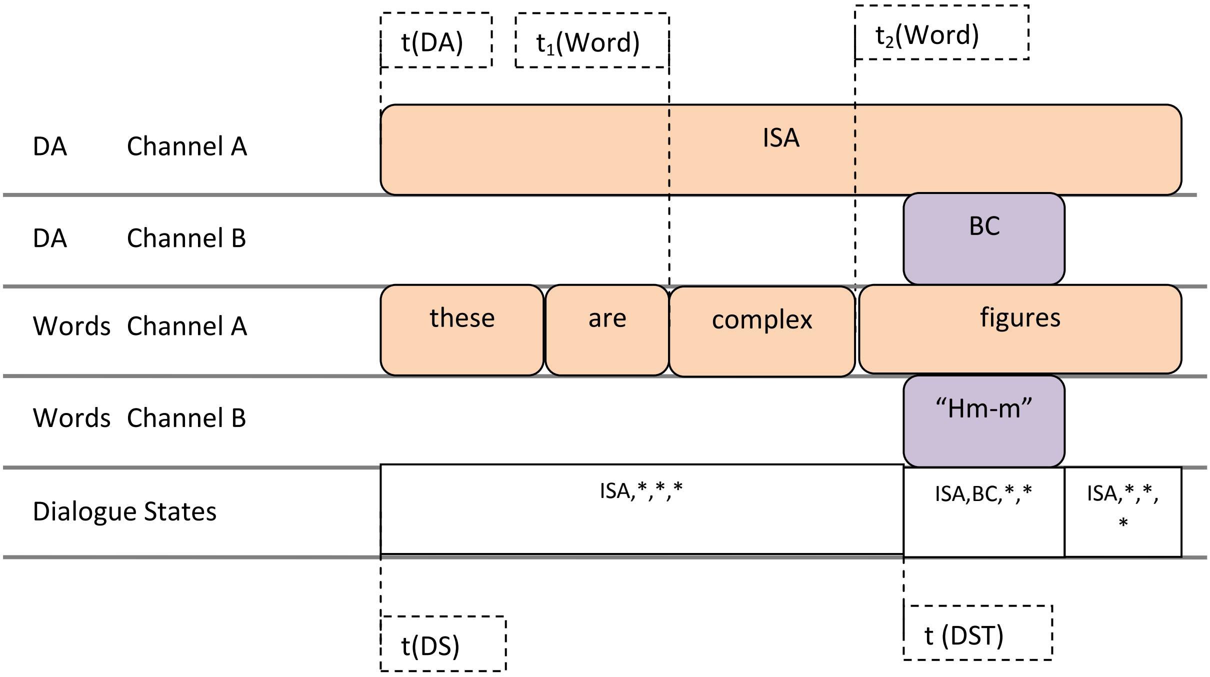


Figure 4.11 Starting time indices: a word, dialogue state or dialogue act. In this example DS and DA are of equal length because there is nothing else in between, they start at the same time index.

In this table “ms” stands for millisecond, t(DST) is moment (accurate to 10 ms) at which in the corpus the actual transition takes place. So this is the end of the audio sample – all audio succeeding this moment occurs, while the backchannel (or other dialogue act) is taking place and should therefore not be used. t(DS) is the moment at which the dialogue state starts, that preceded the DST. t(DA) stands for the starting time of the dialogue act that surrounds the dialogue state. This is also the DA from which the prosody is extracted over the (sub)sample’s duration. t_1(word) is the starting point of the very last word that was uttered before t(DST) and t_2(word) denotes the end time. If t(DST) intersects a word interval (in other words: a word was being uttered at t(DST)), then the previous word is used as ‘last word’. For further clarification, see the figure.

subsample	Name prefix	Start index	End index
DS	ds_	t(DS)	t(DST)
DA	da_	t(DA)	t(DST)
Word	w_	t_1(word)	t_2(word)
Last sec	sec_	t(DST) – 1000ms	t(DST)
Last 500	500_	t(DST) – 500ms	t(DST)
Last 300	300_	t(DST) – 300ms	t(DST)
Last i * 200	It1000_	t(DST) – 1000ms	t(DST) – 800ms
	It800_	t(DST) – 800ms	t(DST) – 600ms
	it600_	t(DST) – 600ms	t(DST) – 400ms
	It400_	t(DST) – 400ms	t(DST) – 200ms
	It200_	t(DST) – 200ms	t(DST)

Table 4.3 Used names and intervals

The features that are extracted can be of two types: ‘context’ and ‘prosodic’ and fall into two different categories: ‘long’ and ‘short’, referring to their interval length. In the experiment we often use a combination of long and short features, where the short ones apply to a small interval close to the dialogue state transition. Their values are compared against the values of corresponding features, that were extracted over a longer period in time. The first four features are contextual features, extracted from the contextual dialogue act (or possibly another longer period), that surrounds the dialogue state. They are thus the context features for the long interval. This set is included in every experiment.

name	Short description
Dag_label	The group name to which this DA’s type belongs (ISA, EL, …)
Word_amnt	The amount of separate words that this DA contains up to the moment of t(DST) in the example this would amount to 3.
Da_dur	The duration up to the transition: t(DST) – t(DA)
Da_w_l_avg	The average length of each word in this DA, measured over the start to the last completed worde before t(DST), so (t(DA) - t_2(word)) / word_amnt

Table 4.4 The contextual act feature set: always included

This is followed by a list that make up the prosodic feature collection for the long interval. In this list the features are measured over DA, thus over t(DA) to t(DST)

name	Short description
da_p_mean	The mean F0 (pitch)
da_p_sd	The standard deviation in F0 (pitch)
da_vfrms	The total number of voiced frames
da_vf_r	The ratio of voiced frames over total frames (voiced + unvoiced)
da_vf_ps	The average Speech rate: voiced frames / sec
da_i_mean	The mean intensity (dB)
da_i_sd	The standard deviation of intensity (dB)

Table 4.5 The dialogue act (long) feature set

Next up are the listings of the prosodic features for the short intervals. Many of them contain the difference between the extraction from the subsample and the DA sample. Each feature name that uses the postfix “_dif” or “_d” denotes the remainder of the extracted value of corresponding subsample minus the value of the same feature, extracted from the DA sample:

Subsample_Feature_dif = feature_value(short interval) – feature_value(long interval).

The list for the subsamples “word”, “1000ms”, “500ms” and “300ms” contain several of these features. The numbers represent the unit this feature is measured in (dB, Hertz, seconds, etc) and they are not made absolute, so they may be negative values. Most classifiers can handle negative integer values. The next list belongs to the last word.

name	Short description
w_dur	The length of the last word (= sample length)
w_pau_dur	The duration of pause after end of last word: t(DST) – t_2(word)
w_i_m_dif	The difference in mean intensity
w_p_m_dif	The difference in mean pitch
w_p_vf_dif	The difference in frequenty of voiced frames (total)
w_avg_l_dif	The difference in length from average word length
w_i_RFC	RFC intensity slope
w_p_RFC	RFC pitch slope

Table 4.6 The word (short) feature set

The term RFC is used from the Rise Fall Continue term. It can be used here as Rise, Fall, Continuous, because it is used to indicate if a part of a slope rises, falls, or is relatively flat (continuous). For each feature, that has an RFC calculation, the interval is divided in four parts, as Figure 4.12 shows. For each of these parts the feature is calculated: this can result in a pitch or intensity value. Then these values v_1 to v_4 are compared to each other, where a discrete value denotes the difference between v_i and v_{i+1}. The result of each comparison is an “R” whenever v_{i+1} was more than a preset threshold variable higher than v_i; the comparison receives an “L” if it is the other way around and an “C” in neither is true. These three evaluations lead to three letters denoting the value for this slope. Possible values are { $s_1s_2s_3$ | s_i Є {R,F,C}} resulting in 3^3 different possibilities.

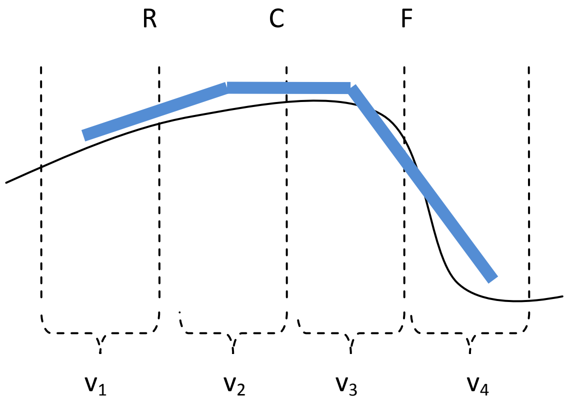


Figure 4.12 RFC representation of a slope

The following list belongs to the subsections corresponding to the last 300ms, 500ms, 1000ms of the DA before the transition occurred. Like the 'last word' feature set, these are extracted from the short interval and belong to the prosodic category. The name "{300,500,s}_featurename" is used to denote all of them in this list (their meaning is straightforward). In the experiment, they are named uniquely and are extracted over the remaining interval before t(DST) according to their names.

name	Short description
{300,500,s}_i_m_diff	The difference in mean intensity
{300,500,s}_p_m_diff	The difference in mean pitch
{300,500,s}_p_vf_diff	The difference in frequency of voiced frames
{300,500,s}_w_i_RFC	RFC intensity slope, same as with the word set
{300,500,s}_w_p_RFC	RFC pitch slope, same as with the word set

Table 4.7 The short sample feature set

The last list in the prosodic category, extracted from a short interval is the 'delta' feature set. This collection consists of 5 different value for each attribute, taken over 5 sequential subsamples, each with a length of 200 ms, delivering average values. The values for these features are the difference between the mean extraction of current subsample minus the mean extraction of the previous section, where the first subsample is starting at 1000 ms before t(DST) – without any subtractions – and the last at 200 ms before t(DST). So for example

It1000_p_d = average pitch over((t(DST) – 1000) to (t(DST) – 800))

It800_p_d = average pitch over((t(DST) – 800) to (t(DST) – 600)) minus it1000_p_d

All attributes within the 'delta' feature set describe the slope of the item, corresponding to the feature name, over the last second of audio, before the transition. The values are, similar to the RFC, an average over the subsample's interval and expressed in continuous numbers, in contrast to the RFC, that used discrete denominators. It is possible to use a subset of these in classification. They are all extracted, so that it1000 remains the starting number and the following attributes retain the difference depicting value.

name	Short description
{it1000, it800, it600, it400, it200,s} _i_mean_d	The difference in mean intensity between this item and the previous, where it1000 precedes it800. it1000_i_mean_d is used for the difference (numeric value) between the mean intensity obtained from the interval inbetween 1000 and 800 ms before t(DST) and the mean value over the longer interval: da_i_ mean. The same goes for the 'p' (pitch) 'sd' (standard deviation), and vfr (voiced frames frequency)
{it1000, it800, it600, it400, it200,s} _i_sd_d	
{it1000, it800, it600, it400, it200,s} _p_mean_d	
{it1000, it800, it600, it400, it200,s} _p_sd_d	
{it1000, it800, it600, it400, it200,s} _vfr_d	

Table 4.8 The delta feature set

There is one last contextual feature set that caught our eye. We speak of the 'focus of attention' dataset – foa for short. It is extracted at the moment of t(DST) but looks back in time if necessary and is therefore marked as a set that is extracted at a short interval.

For a relatively small amount of meetings, the direction of a participant's gaze was annotated in the F.O.A. modality, as explained in chapter 2. A participant can look at several items; we subdivided these in persons, places or undefined(s). In the list below, the 'speak' nominator is used to indicate the person that will start speaking shortly; elsewhere named as the *next* or *new* speaker.

name	Short description
foa_type	The class of item the gaze is directed at, this can be a person, place or unspecified
foa_target	A label naming the specific target: participant A~D, the table, screen whiteboard or unspecified
foa_at_speak	Whether the next speaker is looked at ("Y" or "N")
foa_at_speak_dur	How long the next speaker was looked at (numeric; 0 if foa_at_target = "N")
foa_e2e	Whether or not the next speaker was looked at *and* was looking back at the previous/current speaker ("Y", "N")
foa_e2e_dur	How long the participants had eye contact (numeric; 0 if foa_e2e = "N")

Table 4.9 The foa feature set

One more feature was extracted from the samples. Because one feature does not make a set, this was included in the focus of attention set. Together they form the multimodal feature set.

name	Short description
Head	A signal that is communicated (or not)

Table 4.10 Head feature set, included with foa from here on

All meetings that received foa-annotations also contain head communication annotations and if we can detect focus of attention in a real scenario, surely head moments would be possible as well, so if this feature can improve overall classification, it may be used with the foa feature set in experiments to com. There is a list of signals that were encoded. We denoted them with the two-letter labels listed below.

Annotaion name	label
Emphatis_signal	SE
Concord_signal	SC
Discord_signal	SD
Deixis_signal	SX
Negative_signal	SN
Other_comm_head	CO
No_comm_heand	CN
Off_camera	OC

Table 4.11 Possible head signals and abreviations

One very different feature set was reported about in chapter 2, where a collection of filters was applied to a collection of samples taken from different lengths of two features: F0 and intensity. (Fuentes, Vera, & Solorio, 2007) Using the same approach, we create the “FilteredSlope”.

We start by taking defining the 3 filters that are applied:

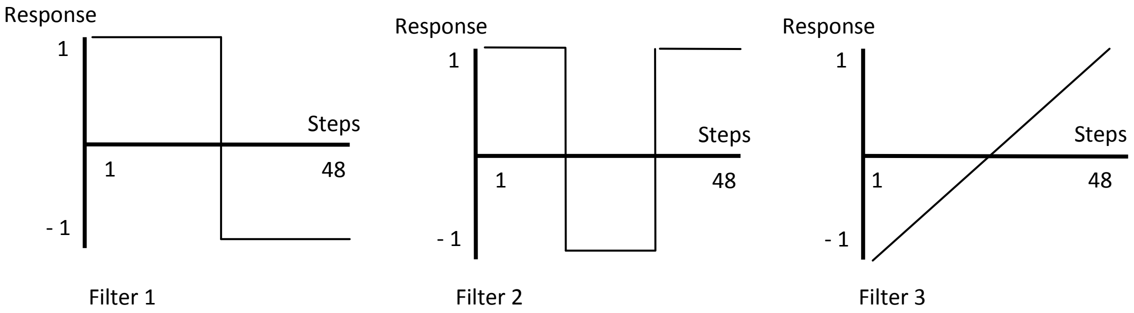


Figure 4.13 Three filters

A filter’s response is either 1 or -1 for filters 1 and 2, depending on the step. We call filter 1 “F_1” and the response is $F_1(s)$ where s is the input step.

For example:	$F_1(1) = 1$.	$F_1(24) = 1$.	$F_1(25) = -1$.	$F_1(48) = -1$.
Likewise	$F_2(1) = 1$.	$F_2(24) = -1$.	$F_2(25) = -1$.	$F_2(48) = 1$.

Filter 3 is a linear interpolated slope between -1 and 1, where $F_3(1) = -1$ and $F_3(48) = 1$.

We explain the algorithm by describing the steps that are taken and applying them directly to an example: extraction of F0 of an interval of 200 ms in length. The dot product from the filters and this sample is calculated in the following manner:

- The 200 ms period is subdivided in a number of intervals, equal to the number of steps as we have in our filter. In this case we used 48. Each interval has a duration of 200/48 ms and the starting and stopping time indices are i * (200/48) and (i +1) * (200/48) respectively, where i runs from 1 to 48. We call these separate intervals $SI_{i.}$. This number ‘48’ is chosen arbitrary: increasing it means more accurate results, but longer required computation time. We recommend using a number dividable by 2 and 3, given the contours of F_1 and F_2.

- Each separate interval SI is sampled on the prosodic feature F0 and the resulting value is multiplied by a filters response at that time step; SI_i * F(i). This is done for each filter separately.
- All products are added together into one numeric value, for each filter. The process is comparable to taking the dot product of 2 vectors of the same size.

Now we have 3 numeric values, one for each filter, representing the F0 slope of 200 milliseconds, multiplied by the filters slope with a step size of 48.

This process is repeated for a collection of 18 different window lengths, starting at 200 milliseconds before the moment of t(DST) and increasing with 100 ms steps to a maximum of 2 seconds before t(DST). The same is done for the intensity slope. This results in a feature set of 2 (features) * 3 (filters) * 18 (sample lengths).

The feature names start with “fs” to denote that they belong to the ‘filtered slope’ feature set, a ‘i' or ‘p’ for intensity and pitch, a number (1, 2 or 3) for the applied filter and a number to indicate the sample length in milliseconds. For example: “fs_p1_200”, or “fs_i2_1800”.

The essential approach was adopted from a study in which a comparable set of feature was taken and an effort was made to detect the end of utterances. Although this is an entirely different classification problem, the results that were obtain lead to believe that this feature set can be used in our task as well.

We will elaborate on each feature set’s performance in the next chapter. In a few sections it reports on the process of experimenting, evaluating intermediary results and making choices based on them. It resembles the larger (relevant) part of the experiments that were conducted and the result that were obtained.

5 Processing results

As explained, the goal is to find and evaluate the prosodic features that perform well, in the sense that they help the classification process that distinguishes moments preceding backchannels from other moments based on the utterance characteristics. At this point an experiment can be seen as a very large function with a number of arguments, that were explained in the previous chapter:

- The collection of occurrences that make up the input dataset
- The collection of features that is extracted from each sample
- The collection of classifiers that is experimented with

Through thorough analysis of different combination of features, taken from different sample lengths, results become more accurate and valuable. There is however a limit on the available computational space and time complexity. During the experimentation phase it became clear that the right balance is needed between exhaustive experimenting and combining feature sets in classification tasks. So, at one side, we need to vary the parameters enough to make sure that the results do not get stuck at a local optimum in the set of possible outcomes. On the other side, trying every combination of feature set and window length is not feasible.

This chapter reports on what was experimented with and what findings were the result. It does so, not by listing a 4-dimensional table (dataset, sample length, feature set, classifiers) with results, but by establishing what choices can best be made for these dimensions, in a linear manner. First we report on experiments with a varying window length, then we compare feature set performances and finally the best training / classification algorithm is chosen.

5.1 Varying window length

The long window

Extractions are done on a sample of the recording of a participant over the whole meeting. Getting the time indices of this sample right is perhaps the most important issue and there are many options. Chapter 2 explained that rises and drops in pitch and intensity should be well explored as they are used by many and are often found to be useful. The choice is made to extract two samples for each item in our dataset, so that a comparison can be made between a short sample, that is extracted moments before the DST and a long sample, that stretches further back in time. Sample lengths will be referred to as 'intervals' and we shall first discuss the long variant.

The DST's that were extracted from the AMI corpus are marked by 1 time index; the moment at which one dialogue state transits to the next: at the bottom of Figure 5.1, this is denoted by "t(DST)". It marks the end point of any sample, because at this point in time one of the participants started a dialogue act, while someone else was already talking. Any point beyond this one, lies outside of the scope of what could be measured for this transition in a real(time) situation. We refer to the participant that starts with the new dialog act and causes the DST as the 'new speaker'. In Figure 5.1 this is the participant on channel B. By the 'current speaker', 'active speaker' , or just 'speaker' we mean the participant that has the floor and is the source of the extracted features; channel A in the same figure.

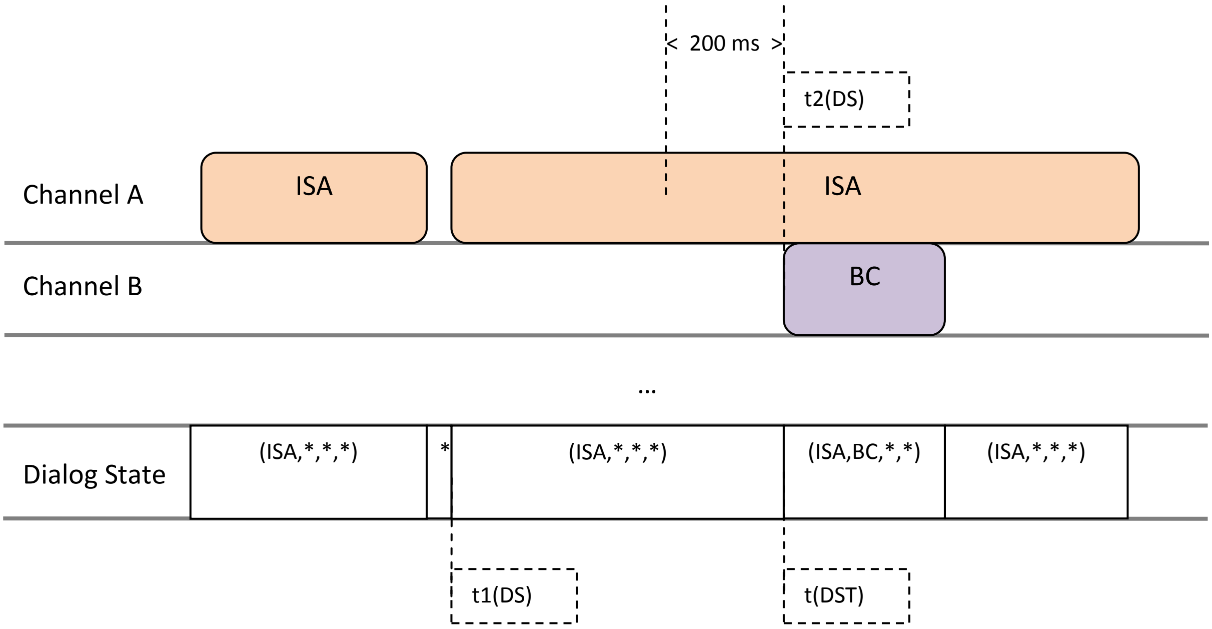


Figure 5.1 The long interval stretches over the preceding dialogue state. In this example it is the same size as the preceeding part of the dialogue act.

For the long sample we can choose between three options:

(1) The interval between the start and end time index of the dialog state (DS) preceding the DST
(2) The interval between the start of the dialogue act (DA), corresponding to the current speaker and the start of the DA uttered by the new speaker.
(3) The interval between the start of the turn of the current speaker and the start of the DA started by the new speaker

(1) Seems an obvious choice. The start and end index of this interval are marked in by t1(DS) and t2(DS), respectively. A result from choosing these indices for the long interval proved unsuccessful because it was too short on many occasions. The extraction procedure handles errors by the marking and removal of occasions of which a feature set could not be extracted for any reason. In early experiments, many items were removed from the dataset in the cleaning process, as table 5.1 shows.

Collection	Total DST	Error producing samples	Accepted items
ES2002 ~ ES2016	6758	2125	4633
IS1000 ~ IS1009	4256	1248	3008
TS3003 ~ TS3012	4050	1286	2764

Table 5.1 Dataset specific accepted items

The problem was traced back to the long interval being very short on many occasions. This happens when several participants start at approximately the same time: for each *n* short dialogue acts, given by any number of participants, causing rapidly following stats/stops, there can be *n -1* items that did not have at least 200 milliseconds of uninterrupted speech needed that has been set as a minimum length requirement. Figure 5.2 shows a situation that would produce a long sample of too little time: there is another dialog act (type Elicit) at another channel (participant C), right in front of the backchannel. This new dialogue act changes the dialogue state, just like any other would. Compared to figure 5.1 we now get a new dialogue state, starting at the EL-type. Now the starting time of the BC preceding interval t1(DS) is set at the start of this new DS, creating a much shorter interval t1(DS) – t2(DS).

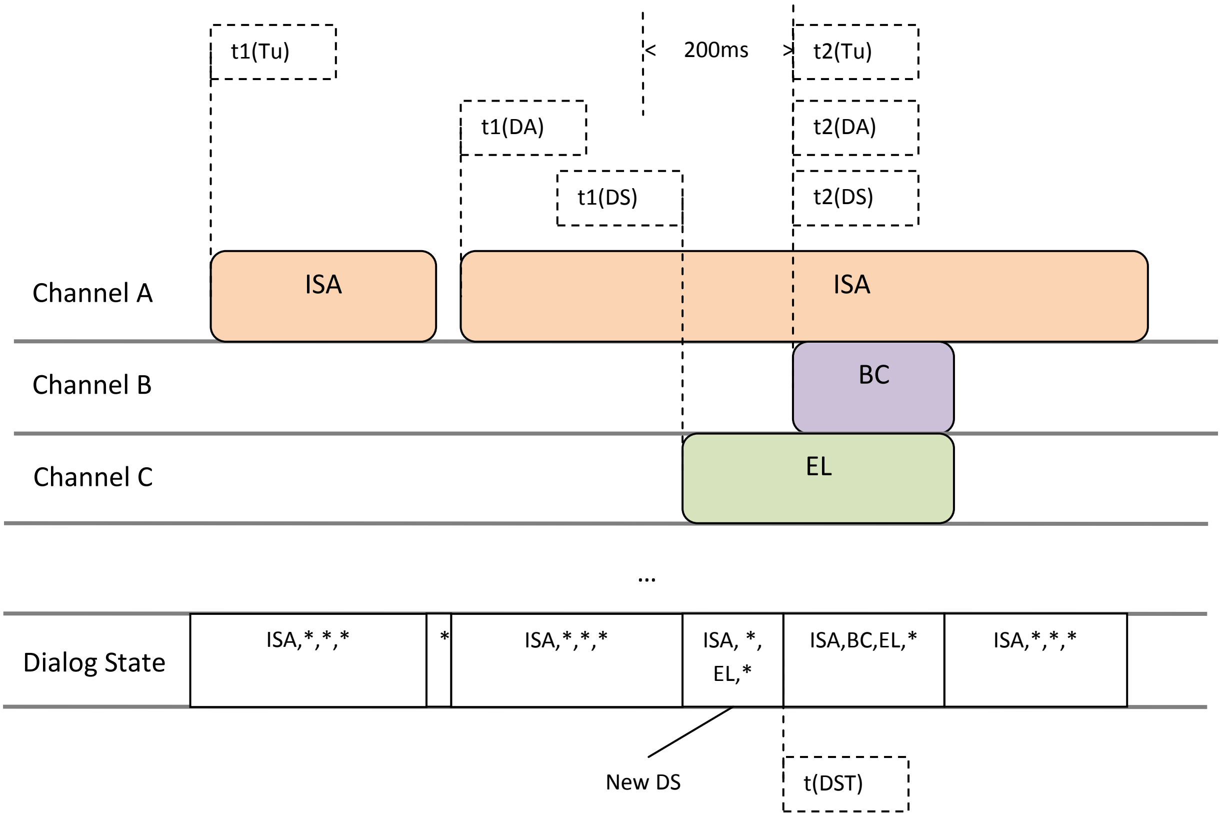


Figure 5.2 The new long (turn) interval stretches further back than the preceding dialogue acts if the silence between these is less than 250 milliseconds

To prevent a sample from getting too short, the choice was made to extend the start back in time to the start of the new dialogue act: choice (2) with time indices: t1(DA) – t2(DA), in figure 5.2.

The third option mentioned above will be discussed shortly. First the feature set extracted from this long interval is combined to a set that is extracted from the short interval, so the first results can be created and evaluated.

5.1.2 short

The second feature set is used for comparison to the corresponding features over the entire dialogue act (the long interval). The value for each feature is subtracted with the corresponding value from the long set and the result can be given in discrete or continuous form. Take, for example Table 5.2, showing possible values for the mean pitch feature:

Da_p_mean value	Da_w_mean Value	Printed discrete	Printed continuous
180 Hz	125 Hz	“Less”	- 55

Table 5.2 Mean pitch value example

As with the previous, longer interval, one can also think of any number of possible periods, for the short interval. We choose to compare the performance of these four different ‘prosodic windows’ as the second feature set:

- The last spoken word
- The last 300 millisec of the utterance
- The last 500 millisec of the utterance
- The last 1000 millisec of the utterance

The ending time indices of these intervals are set on the moment of the DST *only* if a word was being outspoken at that exact moment (Praat is very precise…). If not, then the end time is set at the ending time of the last outspoken word. (The period between the end of the word and the start of the new dialogue state will count in the ‘pause’ feature.) In order to measure which set performs best, intermittent results were obtained by using the Weka toolkit on a dataset of extracted feature values. Different datasets were created: one for each of the three collections of meetings, named in chapter 1.2. They are listed again in table 5.2, below.

Collection	name
ES2002 ~ ES2016	ES
IS1000 ~ IS1009	IS
TS3003 ~ TS3012	TS

Table 5.2 Dataset abbreviations

Several different classifiers were tested on these first datasets, giving varying results. We choose to continuously use Weka’s implementation of the J48 decision tree classifier to test performances of different datasets. J48 was chosen because it is one of the best performing classifiers on average and works very fast in comparison to others. It was applied, using a 10-fold-cross-validation training and testing method on the 3 different datasets and their unison. The Table 5.3, below lists the results obtained from the united dataset (ES+IS+TS). In this table, the feature set label indicates which interval was used from which the prosodic set was extracted. “DA + last 300ms” implies that the long interval had the start of the corresponding dialogue act as its t1. Likewise the short interval had t(DST) - 300 ms as its starting time. “DA + all” implies that all the extractions for all 4 separate short intervals were included. For detailed results on the separate datasets (ES, IS, TS), see appendix B1 to B5 in the same order as the feature sets are the table; one appended sheet per line. (Please note the difference between ‘data set’ and ‘feature set’.)

Exp	Feature set	Correct	Precision	Recall	f-Measure
1.1	DA + last word	61.7 %	Ba: 0.61 NBa: 0.624	Ba: 0.639 NBa: 0.595	Ba: 0.624 NBa: 0.609
1.2	DA + last 300 ms	61.8 %	Ba: 0.61 NBa: 0.627	Ba: 0.628 NBa: 0.607	Ba: 0.632 NBa: 0.632
1.3	DA + last 500 ms	61.8 %	Ba: 0.61 NBa: 0.627	BA: 0.646 NBa: 0.589	Ba: 0.628 NBa: 0.607
1.4	DA + last 1000 ms	61.8 %	Ba: 0.61 NBa: 0.627	BA: 0.646 NBa: 0.589	Ba: 0.628 NBa: 0.607
1.5	DA + all	61.9 %	Ba: 0.607 NBa: 0.632	BA: 0.664 NBa: 0.574	Ba: 0.634 NBa: 0.602

Table 5.3 10-fold J48 Performance for short interval variations with the long dialogue act interval

the 3 + 1 different sets were purged of all items that generated errors while being processed. In the praat script these generate output that can be identified later; the item is discarded and the next is analyzed. In total, there were 19971 DST processed, 14282 of which were accepted and 5689 were removed. The accepted items were distributed 7114 / 7168 over the Ba / NBa classes respectively.

The rather large number of unaccepted items is mainly caused by dialogue state transitions that lack enough sampling space in front of the transition. Recall that the dialogue act, rather than the dialogue state was chosen to be sampled for the long interval. If we compare these numbers to the removals that occurred from choosing the shorter interval (see [table 5.x]), we can see that there is an improvement of accepted items. There are however still many unaccepted feature extractions. Possibly, the long interval is still a bit short.

In this experiment (and those to come) a weighed dataset was input into Weka's classifies, creating ideal conditions for classification: approximately the same number of items for both groups. Based on the removal rates of the separate datasets, the parser was adjusted so that a balanced dataset would be created.

The results that were obtained from the first experiment are shockingly similar over the different feature sets, and above all, not very impressing. By using a balanced dataset over 2 different classes, we set the baseline at 50%. ((7158 / (7158+7168) = 49.965) An overall increase between 11 and 12% over that minimum is rather disappointing. The detailed results on precision, recall and f-measure aren't exciting either: there is practically no difference between feature sets . Recall and f-measure score a little higher on the Ba class, precision is a bit higher for the NBa class, on all separate sets. This leaves us with two probabilities:

- There is not much to be learned from pitch and intensity measurements over any part of the recent history of an utterance
- The DA feature set is mainly responsible of the 11.8% increase over baseline.

The first assumption is a bit worrying given the expectations. Remember however, that this was a rather small feature set over relatively greatly varying, short interval. The next experiment attempts to cut down on the losses by increasing the long interval.

5.1.3 longer

There is another choice to explore for the long interval, namely the entire turn up to the moment at which the new speaker started. At the start of the paragraph this was option number (3) and figure 5.2 denotes the time indices for this interval as t1(Tu), t2(Tu) for start and stop respectively. In this case the term 'turn' is slightly inappropriate but it still describes the actual interval the best. With this term, we mean the period of speech that was not interrupted by a period of silence longer than 200 milliseconds. The long interval feature set is now labeled as 'Turn' (see also table 5.4)

The new period is selected for the 'long interval' and the combination of feature sets is repeated in a new experiment. What interests us is the difference in results that is made by lengthening the long interval. In a real (time) situation the selection of this period might be more expensive in terms of computing complexity, because features like mean and standard deviation must be calculated over more values as the time progresses and an interval would get longer (remember Praat's 10 ms sampling method). It would also be simpler to apply in a real-life scenario: there is no need for an algorithm that determines the start of a new dialogue act, but rather the start of speech would be enough.

Again we list the summary of the experiment. Table 5.4, below, shows the results obtained from the combined dataset. The training and classification method was kept the same, as well as the input data set for the prosodic feature extraction and the small interval feature sets. For more detailed results please consult appendix C1 to C5, corresponding to experiment ('Exp' column) 2.1 to 2.5

Exp	Feature set	Correct	Precision	Recall	f-Measure
2.1	Turn + last word	65.1%	Ba: 0.635 NBa: 0.666	BA: 0.628 NBa: 0.672	Ba: 0.632 NBa: 0.669
2.2	Turn + last 300 ms	65.4%	Ba: 0.634 NBa: 0.674	BA: 0.648 NBa: 0.66	Ba: 0.641 NBa: 0.667
2.3	Turn + last 500 ms	65.5%	Ba: 0.638 NBa: 0.669	BA: 0.631 NBa: 0.676	Ba: 0.635 NBa: 0.672
2.4	Turn + last 1000 ms	65.9%	Ba: 0.637 NBa: 0.681	BA: 0.659 NBa: 0.659	Ba: 0.648 NBa: 0.67
2.5	Turn + all	65.4 %	Ba: 0.633 NBa: 0.674	BA: 0.648 NBa: 0.659	Ba: 0.64 NBa: 0.666

Table 5.4 10-fold J48 performance on short interval variations with the long turn interval

The merging and cleaning process produced completely different results. As expected, more items were accepted because we look back in time for preceding dialogue act's with no (or a very small) gap in between. This way less samples would fail data extraction caused by an interval duration of less than 200 milliseconds. The difference between the two selection methods exceeded expectation as in this scenario only 1547 (instead of 5689!) items were removed. So, once more 19971 items were processed, but now 18424 made it to the eventual dataset.

Not only did we produce more results, they are better on average as well – although not very much. Furthermore, there is a small variation in the precision, recall and f-measure columns between the two classes, possibly due to a little shift in the balancing. In experiment series 1, items from the NBa group were discarded more often than those of the Ba group. To create a balanced input set for training, the weighing algorithm was set to compensate for this difference. As a result the NBa group was 9% larger at time of input for the prosodic extraction. In the second series, there is no significant difference in these removal proportions, but in the total removals: this set was reduced with 73%. Obviously this causes a different resulting proportion: for the combined set this is a 8761 / 9663 distribution over the Ba / NBa classes respectively. (see the appendix C1-5) for the separate sets) Note that this imbalance affects the baseline as well; if all items were simply to be classified as being part of the NBa class, the baseline classifier would achieve 52.4% instead of the previous 50.0%.

If we look at the general improvement, it is concluded that the longer – turn – interval clearly performs better that the shorter – dialogue act or dialogue state – and is to be used in next experiments. The classification performance has increased more than the baseline. Furthermore, all scores from the two corresponding tables 5.3 and 5.4 on precision, recall and f-measure are compared for both classes, 28 out of 30 are better.

Although the difference between some did get larger in the performance of different short sample feature sets (w, 300, 500, 1000), it is still too small to clearly state that one performs better than the other. In conclusion, it is surprising to see that – although only by 0.5% – the 'turn + 1000ms' set, outperformed the 'turn + all' set, which also includes the '1000ms' feature set. We shall look in to this in the next experiment.

5.2 Feature sets' results

The previous section focused on the long and short interval feature sets. In the 'last word' and fixed-amount-of-time, short-interval feature sets (300ms, 500ms, 1000ms), the contents mainly consisted of discrete comparisons against the averages of the longer interval, for corresponding features. Although this did not result in a very exiting aftermath, it did provide simple and fast comparisons. It was concluded that the use of a longer interval resulted in better performances. This section will now 'zoom' in on the shorter interval by discussing the result obtained from the 'delta' feature set. The 'long' interval's features, shown in table, will be extracted over an interval ending at t(DST) (like most features) and start either at the start of the speakers turn, or 15 seconds before t(DST), whichever results in the shortest sample. The balance lies between: as much as possible without stepping in front of the speakers turn, but still manageable in terms of time complexity. Because the entire turn is not always the interval for this set and because this set is used in the following experiments, it is now labeled the *base set.* The features are those of tables 4.4 and 4.5, extracted over the interval mentioned above.

As explained in chapter 4, this feature set is also based on the slopes of certain features, like the RFC (rise, fall, continuous) slopes that were used in the other short interval feature sets. However, where RFC was printed with a fixed number of nominators (Le(ss), Mo(re), Co(ntinuous)), the new features are depicted with continuous values. Remember the short example from 5.1.

Da_p_mean value	Da_w_mean Value	Printed discrete	Printed continuous
180 Hz	125 Hz	"Less"	- 55.0

Table 5.4

For experiment series 3, we use the 'turn' based, long interval feature set, concluded to perform the best in the previous section. Since this set produced many more accepted items, the weighed input resulted in a extracted dataset that was a little off balance. To rebalance this, the weights in the parser were adjusted from Ba : NBa = 100 : 109 to Ba : NBa = 100 : 104. As chapter 3 explained, the NBa set is randomly selected from the much larger NBa group. This entails that the input for the NBa group will not be the exact same set of items as for experiment series 1 and 2. However, since the dataset contains more than nine thousand separate items for both groups, this should result in an practically equal (or at least a comparable) dataset.

We experimented with several different lengths of the delta set, combined to the turn-set for the long interval. As chapter 4 explained, the delta set calculates the difference between this step and the previous, where the step size is 200ms and the maximum time in the past is 1 second. Feature values are numeric and “last 400 ms” means that the differences were calculated from the features at t(DST) – 400 to t(DST), where every 200 ms a new set of feature differences is included in the set.

The feature set names in table 5.5, showing the delta results, below, are mean recursively, meaning that the name ‘400ms’ implies that the two separate sets delta(400) and delta(200) were used; so ‘1000ms’ has all five. In the cleaning process, out of 19971 items, 2984 were rejected and 16987 were accepted, distributed over Ba / NBa with a 8461 / 8526 spreading.

Exp	Feature set	Correct	Precision	Recall	f-Measure
3.1	Base + last 1000 ms	62.9%	Ba: 0.624 NBa: 0.634	BA: 0.633 NBa: 0.626	Ba: 0.633 NBa: 0.626
3.2	Base + last 800 ms	63.3%	Ba: 0.626 NBa: 0.639	BA: 0.649 NBa: 0.615	Ba: 0.638 NBa: 0.627
3.3	Base + last 600 ms	63.4%	Ba: 0.625 NBa: 0.643	BA: 0.66 NBa: 0.608	Ba: 0.642 NBa: 0.625
3.4	Base + last 400 ms	64.0%	Ba: 0.628 NBa: 0.653	BA: 0.677 NBa: 0.602	Ba: 0.652 NBa: 0.627
3.5	Base + last 200 ms	64.5%	Ba: 0.646 NBa: 0.643	BA: 0.634 NBa: 0.655	Ba: 0.64 NBa: 0.649
3.6	Base	64.5%	Ba: 0.648 NBa: 0.642	BA: 0.629 NBa: 0.66	Ba: 0.638 NBa: 0.651

Table 5.5 incrementing short interval length

Once more, the differences are not very large between the different lengths, but is very remarkable that the shorter the slope becomes, the better the results get. This is puzzling, since the second series of experiment held the best results for the feature set ‘turn + 1 second’. The differences between these two series are:

- Series 2 uses simple, discrete values like “More” or “Equal”, whilst series 3 uses continuous, numeric values; surely these can retain more information.
- Series 2 uses two values per feature to indicate the difference between the last second and the entire turn: an absolute (continuous) value and the difference (discrete). Series 3 uses 6 different values: the turn’s average, the average over (-1000 ... -800) and the remaining 4 values denoting the difference between their average, and their previous (the 200ms stepping). Again, per feature, series 3 holds more information.

We take a closer look at the features themselves. The appendices show a list of feature names and numbers, under the classification-scores table (series 1 and 2 (appendix B and C) as well), looking like table 5.6, below. The table lists the top five ranked features

average merit	average rank	attribute
0.07 +- 0.001	1.1 +- 0.3	2 word_amnt
0.069 +- 0.001	1.9 +- 0.3	7 base_vfrms
0.058 +- 0.001	3 +- 0	3 base_dur
0.028 +- 0.001	4 +- 0	1 dag_label
0.018 +- 0.001	5 +- 0	8 base_vf_r

Table 5.6 Feature Top 5, delta set

For the upper line, this means, that – measured over 10 fold cross validation – Weka's attribute selecting algorithm ("InfoGain") with default ranker and default parameters, found feature number 2, "word_amount", to have the most value for experiment 3.1. This algorithm ranks all the features that were used for classification. On average it was ranked 1.1 with a standard deviation of 0.3. For each experiment the ranking is given over the combined (ES+IS+TS) dataset; this is gathered in the respective appendices. If the other rankings are consulted as well, it becomes apparent that the *base feature set* has more overall value than the slope features over any interval in the last second. All experiments of this series list the features in table 5.6 as their top-5 attributes and in the same order as well.

In conclusion the delta feature set did not yield a productive contribution. This leaves us with the distinct feeling that there is little to be gained in the last moments before the dialogue state transition occurred.

5.3 Increasing the resolution

The previous section left us with doubts about the merit of using slope values for classification between the Ba and NBa groups. In this section we use a different approach, by literally looking at a small sample from the dataset. A script was written, that extracts F0 values from the speakers utterance over the last 500 milliseconds before the dialogue state transition, with time steps of 10ms. (This is also the resolution used in the corpus in time indices.) By connection these values we visualized the pitch slopes.

Figure 5.3 Shows the pitch contours extracted from 100 different samples that belonged to the Ba group. The number was chosen as the balance between getting a good distribution and manageable visibility. In order of maximizing the chance to detect a distinctive pattern in the pitch slopes, all samples were randomly picked from one randomly selected person in exactly 1 randomly selected set of meetings (i.e. ES2002a,b,c,d) (any meeting or person could be sampled). On the vertical scale we see the pitch in Hz, on the horizontal there is the time in steps of 10 ms where '-50' indicates t(DST) – 500ms. Pitch values below 50 Hz have been cut off.

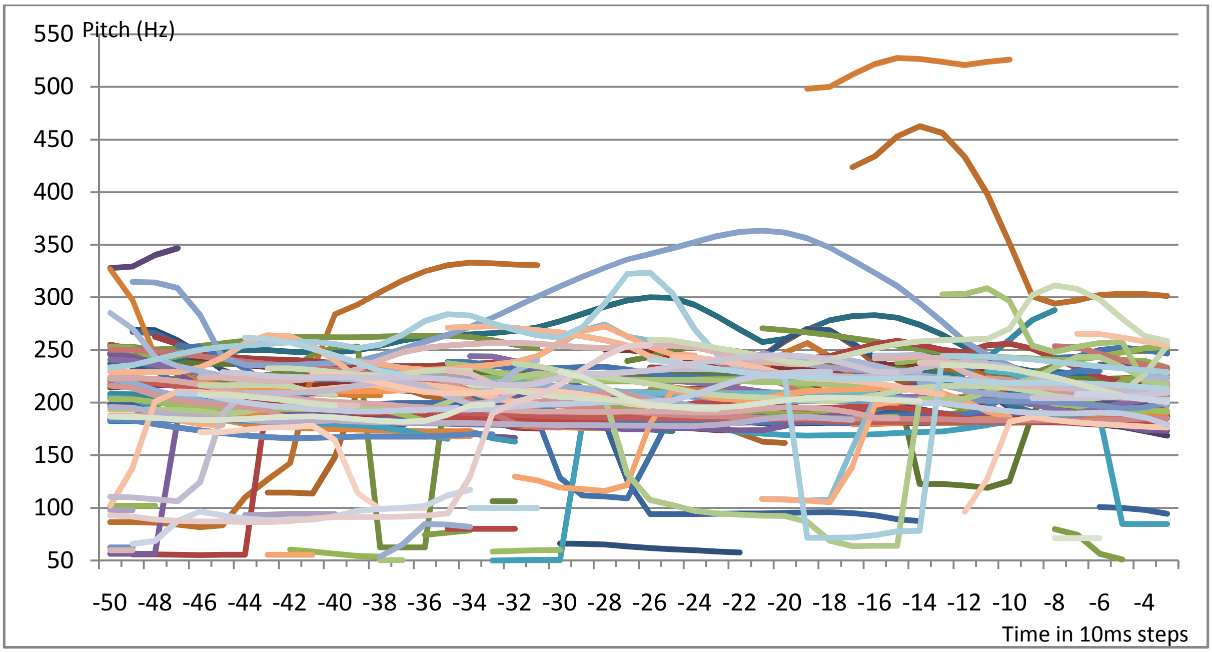


Figure 5.3 Pitch slope showing 100 utterances preceding a backchannel

As we can see, there are a few steep falls and rises. Those jumping to and from regions around 50 Hz indicate starts and stops of utterances, since those sounds are far below the average pitch value of approximately 220 Hz. There are a few pitch slopes that beautifully indicate pitch falls at the end of this sample, but there are so few of them. Sadly, the vast majority of pitch slopes lies in between the 170 and 260 Hz regions.

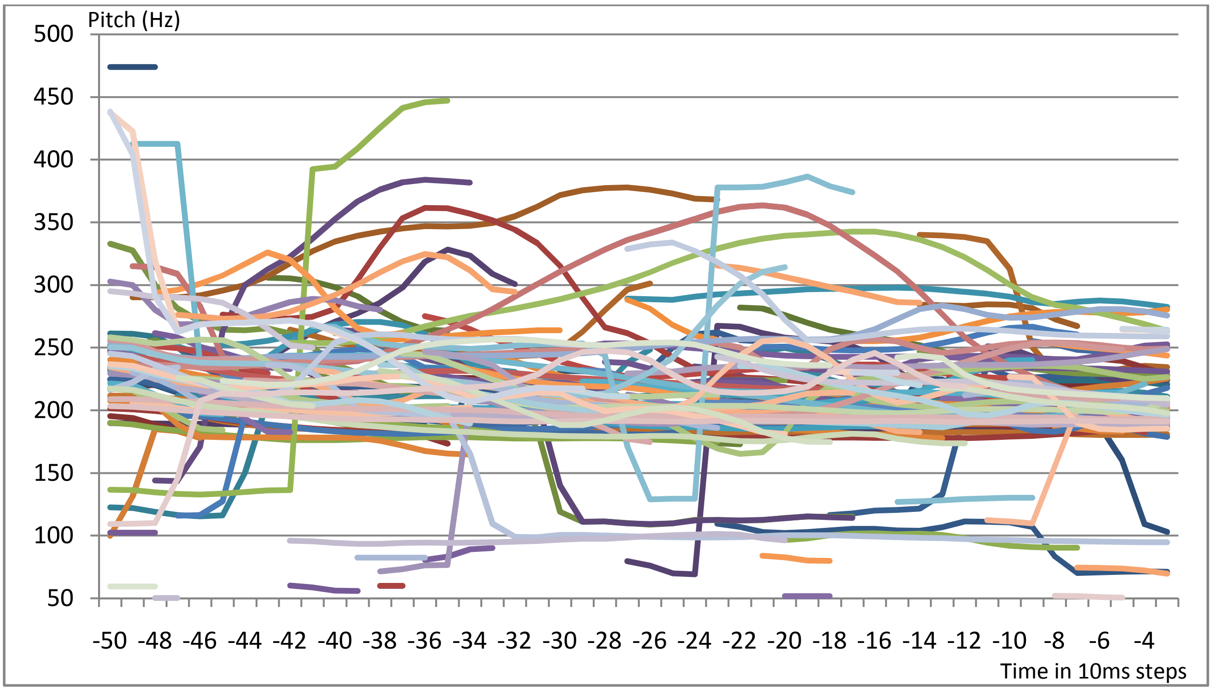


Figure 5.4 Pitch slope showing 100 utterances preceding a non-backchannel

The same was done for 100 samples taken from the other group: NBa. The result is shown in Figure 5.4 These were taken from the same collection of four meetings, from the same person. On first notice there seems to more deviation from the 'gray' midsection, especially the 'excursions' between 250 and 400 Hz occur more in this graph than in the 'Ba' group of Figure 5.3. They seem to be a little higher as well. Still these 'risers' make up for no more than 12, maybe 15 percent of this random selection (of items from the same group). The majority still resides in the midsection.

Because this image only shows the slopes of little over 1 % of the complete dataset (around 9k items for both groups), we cannot conclude on this data. Still we would like to notice that a few items (12 ~ 15% in this case) clearly stands out from the mid section. Furthermore, the pitch slopes appear to have a larger variation in the NBa set than in the set, in the midsection at the same moment before t(DST). Except for the few high value 'strangers' in the latter set, the similarities in these images strengthens the notion that little information can be used from pitch values to distinguish between both groups.

5.4 Filtered slopes

The last feature set that was described in chapter 4, was a set for witch three filters were combined with extractions of varying lengths for F0 and intensity values. The resulting feature set, next to the base set, consists of 18 different sampling lengths to which three filters were applied. This was done for both speech intensity and F0 so we obtained 108 features in the filtered slope set. As the dataset after feature extraction has become of considerable size, the three different (IS, TS, and ES) sections have not been merged, but are evaluated separately in table 5.7 below. The features are *base* + *FS* for each of them and the training and classification method that was applied was J48 with a 10-fold cross evaluation.

Exp	Data set	Correct	Precision	Recall	f-Measure
4.1	ES2002 ~ ES2016	59.8%	Ba: 0.583 NBa: 0.607	Ba: 0.658 NBa: 0.529	Ba: 0.619 NBa: 0.566
4.2	IS1000 ~ IS1009	55.5%	Ba: 0.574 NBa: 0.640	Ba: 0.734 NBa: 0.466	Ba: 0.644 NBa: 0.539
4.3	TS3003 ~ TS3012	58.0%	Ba: 0.540 NBa: 0.586	Ba: 0.731 NBa: 0.380	Ba: 0.622 NBa: 0.461

Table 5.7 10-fold J48 performance on Base + FS set

The input for this experiment was balanced, setting the baseline at 50.0%. Although the classification on general is no better than previously achieved results – that were mostly obtained from considerably less complex features – it *is* interesting to see the IS and TS sets achieve the highest recall values yet, especially since the input was balanced well. Table 5.8 lists the input dataset sizes after balancing.

Set	# items
ES	Ba: 2827 NBa: 2876
IS	Ba: 1713 NBa: 1718
TS	Ba: 1786 NBa: 1776

Table 5.8 base + FS dataset sizes

The corpus that was used in the study from which we adopted this feature set was considerably smaller than ours (little under a thousand items, total), but good results were obtained from it. (Fuentes, Vera, & Solorio, 2007) In the study, a collection of classifiers, primarily consisting of decision tree classifiers was trained and tested on the data. The best performance was obtained from a REP tree classifier, with a recall value on the End-of utterance group of 0.85. The obtained f-measure was 0.841.

After experimentation with different classifiers we obtained high recall values as well. The maximum score of, 0.859 on the Ba group of the IS set was achieved by a DecisionStump decision tree classifier, trained and tested using 10-fold crossvalidation. In all cases the high scores resulted from over classifying on this group: False positives were high as well, resulting in poor overall performance scores. No overall improvement over the Base set could be found.

5.5 Focus of attention

One feature set remains. As described in chapter 3, a part of the AMI corpus was outfitted with annotations on signals that were captured in the multimodal featureset. An experiment was conducted to assess their merit as well. Because this experiment can only use a small part of the corpus, a new set of instances was extracted by the parser. These were combined and resulted in a set containing 1392 items, distributed 698 / 694 items, for Ba and NBa respectively. Of the tested classifiers BayesNet had the best average performances. Table 5.9 shows it's performance used the default 10-fold training and classification method.

Exp	Data set	Correct	Precision	Recall	f-Measure
5.1	Foa	52.9%	Ba: 0.530 NBa: 0.529	Ba: 0.543 NBa: 0.516	Ba: 0.536 NBa: 0.522
5.2	Base + Foa	61.7%	Ba: 0.608 NBa: 0.627	Ba: 0.662 NBa: 0.571	Ba: 0.634 NBa: 0.597
5.3	Base + Foa + word	61.8%	Ba: 0.608 NBa: 0.630	Ba: 0.670 NBa: 0.565	Ba: 0.638 NBa: 0.596
5.4	Base + Foa + word + sec	62.3 %	Ba: 0.612 NBa: 0.636	Ba: 0.675 NBa: 0.571	Ba: 0.642 NBa: 0.601
5.5	Base + word + sec	62.8 %	Ba: 0.614 NBa: 0.641	Ba: 0.683 NBa: 0.568	Ba: 0.647 NBa: 0.602

Table 5.9 10-fold BayesNet performance on determinig FOA's merit

After the initial base + foa set did not achieve a very promising performance, the first tested (word, 300ms, 500ms, 1000ms) set was included. Experimentation afterwards resembles a bad comedy in the sense that performance again peaked, with just the original set. The focus of attention set, like many other seems not able to contribute in our classification task.

5.6 Evaluation

We have tested and evaluated four different feature set combinations. The set best performing achieved an overall classification score of 65.9 % correct, roughly 16% above the 50% baseline of the balanced set. This score is not particularly high, so what can be concluded? Can we say that prosody is use full in out scenario?

The dialogue acts with which the AMI corpus has been outfitted, have all been manually annotated by several different individuals. As a result, there will be inter annotator disagreement, on this body of nearly 117k dialogue acts. Is a backchannel always annotated as one? Is a dialogue act that is not a backchannel always annotated as one of the other class. The short answer of course, is “No.”. A study about backchannel distribution regarding the dialogue acts in the AMI corpus showed that there is a significant disagreement between the dialogue act type “Assess” and backchannels. From a small part of the corpus it was found that in 75% of the backchannel cases the annotators agree on the given label, (Heylen & Op den Akker, 2007).

In an internal rapport for the Department of Human Media Interaction at the University of Twente, it was found from comparing annotations that there was also considerable disagreement between backchannel and other types. This report shows annotations of one meeting and from one of the confusion matrices that are presented it can be concluded that one person annotated 128 backchannels, the other 127. They agreed on 96 items as being backchannels. it that the annotators agreed on 96 dialogue acts out of 128 while comparing. 96/127 and 96/128 concurs with the 75% agreement found in the study, (Op den Akker). The meetings used for this data are also included in our dataset. It seems safe to assume that in the overall set there is also a significant annotator disagreement in our two classes of backchannel of non-backchannel.

From our experiments we found the best performing feature set to be the *base* set, combined with the basic set, extracted from the last second of speech. This set included the list of features that is tabulated in Table 5.10.

In this list the first prefix represents the interval over which was extracted. Here *Base* means that the starting point was chosen at a maximum of 15 seconds before the dialogue state transition occurred, whiteout preceding the start of the speakers turn. If the speaker preceding utterances started later, than this was chosen as a starting point. The ‘w’ prefix indicates that the sample was taken over the last *finished* word in the utterance and the most simple – ‘s’ – is used to indicate a sample length of exactly one second before the time of the DST.

1	Dag_label	The group name to which this DA's type belongs (ISA, EL, …)
2	base_word_ amnt	The amount of separate words that the base sample contains up to the moment of t(DST).
3	base_dur	The total duration of the base sample
4	base_w_l_avg	The average length of each word in the base, measured over the start to the last *finished* word before t(DST)
5	base_p_mean	The mean F0 (pitch) over base
6	base _i_mean	The mean intensity (dB) over base
7	base _vf_r	The ratio of voiced frames over total frames (voiced + unvoiced) in base
8	base _vf_ps	The average Speech rate: voiced (frames / base_dur) in base
9	w_i_m_dif	The difference from base in mean intensity, over last *finished* word
10	w_p_m_dif	The difference from base in mean pitch, over last *finished* word
11	w_p_vf_dif	The difference from base in frequency of voiced frames, “
12	w_i_RFC	RFC intensity slope, “
13	w_p_RFC	RFC pitch slope, “
14	w_dur	The length of the last word (= sample length), “
15	w_avg_l_dif	The difference in length from average word length, “
16	w_pau_dur	The duration of pause after end of last word: t(DST) – t_2(word), “
17	s_i_m_diff	The difference from base in mean intensity, over the last 1000ms
18	s_p_m_diff	The difference from base in mean pitch, “
19	s_p_vf_diff	The difference from base in frequency of voiced frames, “
20	s_w_i_RFC	RFC intensity slope, same as with the word set, “
21	s_w_p_RFC	RFC pitch slope, same as with the word set, “

Table 5.10 Feature list of best performing feature set

Of course, not all these features performed equally well. Weka's GreedyStepwise feature subset evaluator only selects features 1, 2, 3 and 7 from the list in Table 5.10. Ten fold cross validation by the InfoGainAttributeEval algorithm, using the default ranker results in the list of Table 5.11. Once more the durational features perform best.

Average merit	Avg. Rank	Attribute name	Average merit	Avg. Rank	Attribute name
0.066 +- 0.001	1 +- 0	2 word_amnt	0.004 +- 0	10.6 +- 1.43	5 base_p_mean
0.061 +- 0	2 +- 0	3 da_dur	0.004 +- 0	11.5 +- 0.67	19 s_p_vf_dif
0.028 +- 0.001	3 +- 0	1 dag_label	0.003 +- 0	13.1 +- 0.54	18 s_p_m_dif
0.019 +- 0	4.1 +- 0.3	4 base_w_l_avg	0.003 +- 0	14 +- 0.77	17 s_i_m_dif
0.018 +- 0.001	4.9 +- 0.3	7 base_vf_r	0.002 +- 0	15.4 +- 0.66	12 w_i_RFC
0.015 +- 0.001	6 +- 0	8 base_vf_ps	0.002 +- 0	16.2 +- 1.08	10 w_p_m_dif
0.011 +- 0.001	7 +- 0	16 w_pau_dur	0.002 +- 0	17.1 +- 0.94	14 w_dur
0.009 +- 0	8 +- 0	15 w_avg_l_dif	0.002 +- 0	17.7 +- 1.1	13 w_p_RFC
0.004 +- 0	9.8 +- 0.6	20 s_i_RFC	0.002 +- 0	18.4 +- 1.2	6 base_i_mean
0.004 +- 0	10.2 +- 0.98	21 s_p_RFC	0 +- 0	20.1 +- 0.3	9 w_i_m_dif

Table 5.11 Top 20 Ranking of features.

The Experiment used J48 decision tree classifiers, because these could be experimented with very fast. The best results were obtained from a LAD decision tree classifier, using default arguments, based on 10 fold cross validation and testing on the set that encompassed a balanced set of roughly 9.1k items for both groups we achieved 65.9% correct classifications. Where the J48 classifier builds trees, containing hundreds of nodes and leaves, this classifier builds the decision tree for this set using only a small amount of features. Figure 5.5 show a textual representation of this decision tree.

```
: 0,0
|       (1) da_dur < 4.265: -0.302,0.302
|       |       (4) word_amnt < 5.5:              -0.123,   0.123
|       |       (4) word_amnt >= 5.5:              0.138, - 0.138
|       (1) da_dur >= 4.265:                       0.236, - 0.236
|       |       (5) word_amnt < 13.5:             - 0.223,  0.223
|       |       (5) word_amnt >= 13.5:             0.043, - 0.043
|       |       (8) w_pau_dur < 1.112:             0.027, - 0.027
|       |       (8) w_pau_dur >= 1.112:           - 0.302,  0.302
|       |       (10) w_avg_l_dif < 0.423:          0.022, - 0.022
|       |       (10) w_avg_l_dif >= 0.423:        - 0.176,  0.176
|       (2) dag_label = ISA:                       0.099, - 0.099
|       |       (3) w_pau_dur < 0.009:            - 0.433,  0.433
|       |       (3) w_pau_dur >= 0.009:            0.026, - 0.026
|       (2) dag_label != ISA:                     - 0.252,  0.252
|       |       (6) dag_label = R:                -0.161,   0.161
|       |       (6) dag_label != R:                0.126, - 0.126
|       |       (9) da_i_mean < 49.395:           - 0.127,  0.127
|       |       (9) da_i_mean >= 49.395:           0.091, - 0.091
|       (7) s_i_m_dif = Le:                        0.052, - 0.052
|       (7) s_i_m_dif != Le:                      - 0.071,  0.071
Legend: Ba, NBa
```

Figure 5.5 LAD Decision tree's textual

6 Conclusions and further research

After evaluating several sets of features, encompassing not only the sought after prosodic values, but also features applying to the context of the dialogue and – for a relatively small corpus – the gaze of the participants, we found less distinguishing markers than initially expected, given other studies' results on different, but comparable classification tasks.

6.1 Conclusions

Reflecting from overall scores, we feel safe to say that our classification problem did gain use full information from the contextual prosody – although it is not much. the best performing feature set was the set extracted from the entire turn, or as far back as 15 seconds, combined with relatively simple prosodic features, extracted from the last second of speech before the new contribution stated. Next to the amount of words and time, this set consisted of basic prosodic features like the amount of time spend having the floor and – not surprisingly – the pause between the last uttered word and the start of the new dialogue act. There were a few attributes that contributed a little, like mean intensity measured over the entire preceding set of utterances and the difference between this and the intensity of the last second of speech. With a balanced input data set, an overall performance was achieved of 65.9%, roughly 16 % above the balanced baseline. The mentioned features allow for the creation of a predicting model with relative ease. The best performing classifier was the LADtree, from which the decision tree was presented. To our surprise very little could be used from pitch and intensity slopes, nor from periods of low pitch or intensity. This was tested over several intervals and with two different approaches. The same conclusion can be drawn for focus of attention signals. Although they contribute in other classification tasks, they had no merit in this study.

In our opinion there is some merit to the developed method and software. It can provide a list depicting all dialogue states and changes in the meeting's conversation. In any case where the requirements of a specific set of situations can be translated to specific demands on dialogue state or dialogue state transitions, the developed software can be used to supply the experimenter with a subset of applicable items. Whether this is then used for prosodic extraction, or something entirely different does not matter.

6.2 Further research

We can think of two possible ways to improve on our results. One option that seems worthwhile to explore would be inclusion of semantic features. Word sequence probabilities have been found to be use full in end of utterance and end of turn classification tasks. Another option would be to see how the performance would change if only the backchannel and non-backchannel items are selected on which the annotators agree on the class. Although this would result in a smaller dataset, it would be of higher quality. If the creation of a best case scenario would improve on our results, it would strengthen the notion that prosodic attributes can be used in this classification task. It might also be true that human conversation simply allows for a great deal of randomness when it comes to back channeling. Perhaps simulations can shed new light on this matter; simulation wherein human participators must evaluate the 'appropriateness' of any number of randomly generated backchannels of a virtual or embodied agent, addressed by another participant.

References

Atterer, M., Baumann, T., & Schlangen, D. (2008, August). Towards Incremental End-of-Utterance Detection in Dialogue Systems. *Coling 2008* , 11-14.

Boersema, P., & Weenink, D. (2001). Praat, a system for dong phonetics. *Glot International, vol. 5 no. 9/10* , pp. 341-345.

Bosch, t. L., Oostdijk, N., & Ruiter, d. J. (2004). Turn-taking in social talk dialogues: temporal, formal and functional aspects. *Proceedings on 9th Conference Speech and Computer* , pp. 454-461.

Brenier, J., Cer, D., & Jurafsky, D. (2005). The detection of emphatic words using acoustic and lexical features. *9th European Conference on Speech Communication and Technology* , pp. 3297-3300 .

Carletta, J. (2007). Unleashing the killer corpus: Experiences in creating the multi-everything AMI Meeting Corpus. *Language Resources and Evaluation 41 (2)* , pp. 181-190 .

Cathcart, N., Carletta, J., & Klein, E. (2003). A shallow model of backchannel continuers in spoken dialogue. *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL10)* , pp. 51-58 .

De Kok, I., & Heylen, D. (2009). Multimodal End-of-Turn Prediciton in Multi-Party Meetings. *To be presented at ICMI2009 - International Conference of Machine Intelligence.*

Edlund, J., Gustafson, J., Heldner, M., & Hjalmarsson, A. (2008). Towards human-like spoken dialogue systems. *Speech Communication 50 (8-9)* , pp. 630-645 .

Edlund, J., Heldner, M., & Gustafson, J. (2005). Utterance segmentation and turn-taking in spoken dialogue systems. *Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen* , pp. 576–587.

Ferrer, L., Shriberg, E., & Stolcke, A. (2002). Is the speaker done yet? Faster and more accurate end-of-utterance detection using prosody. *In Proceedings of ICSLP 2002* , pp. 2061-2064.

Fuentes, O., Vera, D., & Solorio, T. (2007). A Filter-Based Approach to Detect End-of-Utterances from Prosody in Dialog Systems. *HLT-NAACL - The Association for Computational Linguistics* , pp. 45-48.

Godfrey, J., Holliman, E., & McDaniel, J. (1992). SWITCHBOARD: telephone speech corpus for research and development. *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92 Vol 1.*, (pp. pp.517-520).

Graff, D. (1997). The 1996 Broadcast News speech and language-model corpus. *Proceedings DARPA Speech Recognition Workshop* , pp. 11-14.

Hammerschmidt, K., & Jürgens, U. (2007). Acoustical Correlates of Affective Prosody . *Journal of Voice 21 (5)* , pp. 531-540.

Heylen, D., & Op den Akker, H. (2007). Computing Backchannel Distributions in Multi-Party Conversations. *Proceedings of the ACL Workshop on Embodied Language Processing* , pp. 17-24.

Jonsdottir, G. R., Thorisson, K. R., & Nivel, E. (2008). Learning Smooth, Human-Like Turntaking in Realtime Dialogue. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* , 162-175.

Jurafsky, D., & Martin, J. (2000). *Speech and Language Processing. An introduction to Natural Language Processing, Computational Linguistics and Speech Recognition.* New Jersey: Prentice Hall.

Kolář, J., Shriberg, E., & Liu, Y. (2006). Using prosody for automatic sentence segmentation of multi-party meetings . *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 4188 LNCS* , pp. 629-636 .

Lerner, G. H. (2002). Turn-sharing: the choral co-production of talk-in-interaction. *The Language of Turn and Sequence.* , pp. 225–256.

Magnusson, M. (2000). Discovering hidden time patterns in behavior: T-patterns and their detection. *Behavior Research Methods, Instruments, and Computers 32 (1)* , pp. 93-110.

McCowan, I., Carletta, J., Kraaij, W., Ashby, S., Bourban, S., Flynn, M., et al. (2005). The AMI Meeting Corpus. *Proceedings of the 5th International Conference on Methods and Techniques in Behavioral Research* .

Morency, L.-P., De Kok, I., & Gratch, J. (2008). Predicting listener backchannels: A probabilistic multimodal approach. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 5208 LNAI* , pp. 176-190.

Nishimura, R., Kitaoka, N., & Nakagawa, S. (2007). A spoken dialog system for chat-like conversations considering response timing . *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 4629 LNAI* , pp. 599-606 .

O'Connell, D., Kowal, S., & Kaltenbacher, E. (1990). Turn-taking: A critical analysis of the research tradition. *Journal of Psycholinguistic Research 19 (6)* , pp. 345-373.

Op den Akker, H. (n.d.). Meeting IS1003d - with annotator agreement analysis of AMI dialogue act and addressing annotation. *Internal Report - Department of Human Media Interaction, University of Twente* .

Pesarin, A., Cristani, M., Murino, V., Drioli, C., Perina, A., & Tavano, A. (2008). A statistical signature for automatic dialogue classification. *19th International Conference on Pattern Recognition, ICPR 2008* , art. no. 4761075.

Petukhova, V., & Bunt, H. (2009). Who's next? Speaker-selection mechanisms in multiparty dialogue. *Proceedings of DiaHolmia, 2009 Workshop on the semantics and pragmatics of dialogue* , pp. 19-26.

Schegloff, E., Sacks, H., & Jefferson, G. (1974). A simplest systematics for the organization of turn-taking for conversation. *Language 50* , pp. 696–735.

Schlangen, D. (2006). From reaction to prediction. Experiments with computational models of turn-taking. *Proceedings of Interspeech 2006, Panel on Prosody of Dialogue Acts and Turn-Taking - ICSLP 5* , 2010-2013.

Shriberg, E., Stolcke, A., Hakkani-Tür, D., & Tür, G. (2000). Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication 32 (1)* , pp. 127-154 .

Tommassen, P. (2007). Classification of Meeting Activities. *Internal Report - Department of Human Media Interaction, University of Twente* .

Ward, N., & Tsukahara, W. (2000). Prosodic features which cue back-channel responses in English and Japanese . *Journal of Pragmatics 32 (8)* , pp. 1177-1207.

Wichmann, A., & Caspers., J. (2001). Melodic cues to turn-taking in English: Evidence from perception. *Proc. of SIGdial Workshop on Discourse and Dialogue* .

Witten, I., & Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques* (2 ed.). San Francisco: Morgan Kaufmann.

Zhang, T., Hasegawa-Johnson, M., & Levinson, S. (2006). Cognitive state classification in a spoken tutorial dialogue system. *Speech Communication 48 (6)* , pp. 616-632 .

Appendix A1 Dialogue state distribution > 1 prom

Distribution: dialog-state
Different item count: 2849
Total item count: 182320
Relative amount scale: 10.0 = 1%
Absolute threshold: 0
Minimum threshold (promille): 1.0

Item name	#	Relative amount
[*,*,*,*]	24003	131.65314
[*,*,*,11]	559	3.0660377
[*,*,*,12]	298	1.6344888
[*,*,*,14]	334	1.8319439
[*,*,*,16]	437	2.3968847
[*,*,*,1]	559	3.0660377
[*,*,*,2]	1249	6.850592
[*,*,*,3]	1798	9.861781
[*,*,*,4]	7757	42.546074
[*,*,*,5]	867	4.7553754
[*,*,*,6]	2203	12.083151
[*,*,*,7]	365	2.0019746
[*,*,*,9]	3623	19.871655
[*,*,1,*]	553	3.0331285
[*,*,1,4]	394	2.1610355
[*,*,11,*]	351	1.9251865
[*,*,12,*]	222	1.2176393
[*,*,14,*]	318	1.744186
[*,*,16,*]	350	1.9197016
[*,*,2,*]	1335	7.3222904
[*,*,3,*]	1990	10.914875
[*,*,3,4]	403	2.2103994
[*,*,3,9]	273	1.4973673
[*,*,4,*]	8421	46.188023
[*,*,4,1]	392	2.150066
[*,*,4,3]	401	2.1994295
[*,*,4,4]	339	1.8593681
[*,*,4,9]	394	2.1610355
[*,*,5,*]	735	4.0313735
[*,*,6,*]	2202	12.077665
[*,*,7,*]	315	1.7277315
[*,*,9,*]	3507	19.23541
[*,*,9,3]	205	1.1243967
[*,*,9,4]	372	2.0403686
[*,*,9,9]	484	2.654673
[*,1,*,*]	636	3.488372
[*,1,*,4]	376	2.062308
[*,1,4,*]	409	2.2433085
[*,11,*,*]	433	2.3749452
[*,12,*,*]	286	1.5686705
[*,14,*,*]	347	1.903247
[*,16,*,*]	369	2.023914
[*,2,*,*]	1395	7.651382
[*,3,*,*]	2030	11.13427
[*,3,*,4]	370	2.029399
[*,3,*,9]	241	1.3218517
[*,3,3,*]	210	1.151821
[*,3,4,*]	411	2.2542782
[*,3,9,*]	262	1.4370338
[*,4,*,*]	8447	46.330627
[*,4,*,1]	426	2.3365512
[*,4,*,3]	317	1.7387012
[*,4,*,4]	321	1.7606406
[*,4,*,9]	376	2.062308
[*,4,1,*]	447	2.451733
[*,4,3,*]	462	2.534006
[*,4,4,*]	485	2.660158
[*,4,9,*]	411	2.2542782
[*,5,*,*]	904	4.958315
[*,6,*,*]	2625	14.397762
[*,6,9,*]	183	1.0037297
[*,7,*,*]	310	1.7003071
[*,9,*,*]	3709	20.343353
[*,9,*,3]	212	1.1627907
[*,9,*,4]	339	1.8593681
[*,9,*,9]	551	3.0221589
[*,9,1,*]	192	1.0530934
[*,9,3,*]	279	1.5302764
[*,9,4,*]	325	1.7825801
[*,9,9,*]	478	2.621764
[1,*,*,*]	700	3.8394032
[1,*,*,4]	545	2.9892497
[1,*,*,9]	192	1.0530934
[1,*,4,*]	588	3.2250986
[1,*,9,*]	188	1.031154
[1,4,*,*]	605	3.3183415
[1,9,*,*]	198	1.0860026
[11,*,*,*]	829	4.5469503
[12,*,*,*]	441	2.418824
[14,*,*,*]	714	3.9161913
[16,*,*,*]	538	2.9508557
[2,*,*,*]	2693	14.770733
[3,*,*,*]	2558	14.030276
[3,*,*,3]	183	1.0037297
[3,*,*,4]	398	2.182975
[3,*,*,9]	278	1.5247916
[3,*,3,*]	201	1.1024572
[3,*,4,*]	445	2.4407635
[3,*,9,*]	261	1.431549
[3,3,*,*]	194	1.0640632
[3,4,*,*]	478	2.621764
[3,9,*,*]	259	1.4205792
[4,*,*,*]	11069	60.711937
[4,*,*,1]	563	3.0879772
[4,*,*,3]	521	2.857613
[4,*,*,4]	455	2.4956121
[4,*,*,9]	504	2.7643704
[4,*,1,*]	566	3.1044319
[4,*,3,*]	493	2.704037
[4,*,4,*]	489	2.6820974
[4,*,9,*]	474	2.5998244
[4,1,*,*]	609	3.3402808
[4,3,*,*]	541	2.9673102
[4,4,*,*]	491	2.693067
[4,9,*,*]	536	2.9398859
[5,*,*,*]	1586	8.698991
[5,*,*,4]	236	1.2944274
[5,*,4,*]	246	1.349276
[5,4,*,*]	242	1.3273365
[6,*,*,*]	3511	19.257349
[6,*,*,9]	251	1.3767003
[6,*,9,*]	217	1.190215
[6,1,*,*]	183	1.0037297
[6,3,*,*]	193	1.0585784
[6,9,*,*]	261	1.431549
[7,*,*,*]	541	2.9673102
[8,*,*,*]	346	1.8977622
[9,*,*,*]	4720	25.888548
[9,*,*,3]	275	1.508337
[9,*,*,4]	444	2.4352787
[9,*,*,9]	655	3.5925844
[9,*,1,*]	189	1.0366389
[9,*,3,*]	316	1.7332163
[9,*,4,*]	472	2.5888548
[9,*,9,*]	554	3.0386133
[9,1,*,*]	206	1.1298815
[9,3,*,*]	320	1.7551558
[9,4,*,*]	472	2.5888548
[9,6,*,*]	198	1.0860026
[9,9,*,*]	611	3.3512506

Appendix A2 DST distribution > 1 prom

Distribution: dialog-state-transition
Different item count: 19.541
Total item count: 182.182
Relative amount scale: 10.0 = 1%
Absolute threshold: 0
Minimum threshold (promille): 1.0

Item name	#	Relative amount
[*,*,*,*] -> [*,*,*,16]	247	1.3547609
[*,*,*,*] -> [*,*,*,1]	384	2.1061869
[*,*,*,*] -> [*,*,*,2]	552	3.0276437
[*,*,*,*] -> [*,*,*,3]	650	3.5651603
[*,*,*,*] -> [*,*,*,4]	1347	7.3881087
[*,*,*,*] -> [*,*,*,5]	251	1.3767003
[*,*,*,*] -> [*,*,*,6]	360	1.9745502
[*,*,*,*] -> [*,*,*,9]	1174	6.4392276
[*,*,*,*] -> [*,*,1,*]	367	2.0129442
[*,*,*,*] -> [*,*,2,*]	534	2.9289162
[*,*,*,*] -> [*,*,3,*]	579	3.175735
[*,*,*,*] -> [*,*,4,*]	1367	7.497806
[*,*,*,*] -> [*,*,5,*]	222	1.2176393
[*,*,*,*] -> [*,*,6,*]	371	2.0348837
[*,*,*,*] -> [*,*,9,*]	1132	6.2088637
[*,*,*,*] -> [*,1,*,*]	429	2.3530056
[*,*,*,*] -> [*,16,*,*]	188	1.031154
[*,*,*,*] -> [*,2,*,*]	611	3.3512506
[*,*,*,*] -> [*,3,*,*]	675	3.7022817
[*,*,*,*] -> [*,4,*,*]	1368	7.503291
[*,*,*,*] -> [*,5,*,*]	252	1.3821852
[*,*,*,*] -> [*,6,*,*]	420	2.303642
[*,*,*,*] -> [*,9,*,*]	1124	6.1649847
[*,*,*,*] -> [1,*,*,*]	462	2.534006
[*,*,*,*] -> [11,*,*,*]	209	1.1463361
[*,*,*,*] -> [12,*,*,*]	225	1.2340939
[*,*,*,*] -> [14,*,*,*]	203	1.1134269
[*,*,*,*] -> [16,*,*,*]	280	1.5357614
[*,*,*,*] -> [2,*,*,*]	1171	6.4227734
[*,*,*,*] -> [3,*,*,*]	785	4.3056164
[*,*,*,*] -> [4,*,*,*]	1886	10.344449
[*,*,*,*] -> [5,*,*,*]	443	2.4297938
[*,*,*,*] -> [6,*,*,*]	604	3.3128564
[*,*,*,*] -> [9,*,*,*]	1434	7.8652916
[*,*,*,11] -> [*,*,*,*]	220	1.2066696
[*,*,*,16] -> [*,*,*,*]	240	1.3163668
[*,*,*,1] -> [*,*,*,*]	356	1.9526109
[*,*,*,2] -> [*,*,*,*]	218	1.1956998
[*,*,*,2] -> [*,*,*,4]	396	2.1720052
[*,*,*,3] -> [*,*,*,*]	647	3.5487056
[*,*,*,4] -> [*,*,*,*]	1520	8.33699
[*,*,*,4] -> [*,*,*,2]	201	1.1024572
[*,*,*,4] -> [*,*,*,3]	222	1.2176393
[*,*,*,4] -> [*,*,*,4]	2179	11.951514
[*,*,*,4] -> [*,*,*,6]	313	1.7167617
[*,*,*,4] -> [*,*,*,9]	269	1.4754279
[*,*,*,4] -> [*,*,1,4]	282	1.546731
[*,*,*,4] -> [*,1,*,4]	269	1.4754279
[*,*,*,4] -> [1,*,*,4]	421	2.3091269
[*,*,*,4] -> [9,*,*,4]	215	1.1792452
[*,*,*,5] -> [*,*,*,*]	289	1.5851251
[*,*,*,6] -> [*,*,*,*]	521	2.857613
[*,*,*,6] -> [*,*,*,4]	256	1.4041246
[*,*,*,6] -> [*,*,*,6]	296	1.6235191
[*,*,*,9] -> [*,*,*,*]	1091	5.983984
[*,*,*,9] -> [*,*,*,4]	393	2.1555507
[*,*,*,9] -> [*,*,*,9]	266	1.4589733
[*,*,*,9] -> [9,*,*,9]	185	1.0146995
[*,*,1,*] -> [*,*,*,*]	340	1.864853
[*,*,1,4] -> [*,*,*,4]	271	1.4863975
[*,*,2,*] -> [*,*,*,*]	194	1.0640632
[*,*,2,*] -> [*,*,4,*]	433	2.3749452
[*,*,3,*] -> [*,*,*,*]	616	3.3786747
[*,*,3,*] -> [*,*,4,*]	242	1.3273365
[*,*,3,4] -> [*,*,*,4]	210	1.151821
[*,*,4,*] -> [*,*,*,*]	1559	8.5508995
[*,*,4,*] -> [*,*,2,*]	231	1.267003
[*,*,4,*] -> [*,*,3,*]	308	1.6893374
[*,*,4,*] -> [*,*,4,*]	2427	13.31176
[*,*,4,*] -> [*,*,4,1]	308	1.6893374
[*,*,4,*] -> [*,*,4,3]	193	1.0585784
[*,*,4,*] -> [*,*,4,9]	203	1.1134269
[*,*,4,*] -> [*,*,6,*]	289	1.5851251
[*,*,4,*] -> [*,*,9,*]	249	1.3657305
[*,*,4,*] -> [*,1,4,*]	301	1.6509434
[*,*,4,*] -> [1,*,4,*]	457	2.5065818
[*,*,4,*] -> [9,*,4,*]	248	1.3602457
[*,*,4,1] -> [*,*,4,*]	279	1.5302764
[*,*,4,3] -> [*,*,4,*]	234	1.2834576
[*,*,5,*] -> [*,*,*,*]	276	1.5138218
[*,*,6,*] -> [*,*,*,*]	463	2.539491
[*,*,6,*] -> [*,*,4,*]	269	1.4754279
[*,*,6,*] -> [*,*,6,*]	288	1.5796402
[*,*,9,*] -> [*,*,*,*]	972	5.3312855
[*,*,9,*] -> [*,*,4,*]	441	2.418824
[*,*,9,*] -> [*,*,9,*]	238	1.3053972
[*,1,*,*] -> [*,*,*,*]	428	2.3475208
[*,1,*,4] -> [*,*,*,4]	249	1.3657305
[*,1,4,*] -> [*,*,4,*]	283	1.5522159
[*,16,*,*] -> [*,*,*,*]	185	1.0146995
[*,2,*,*] -> [*,*,*,*]	258	1.4150944
[*,2,*,*] -> [*,4,*,*]	426	2.3365512
[*,3,*,*] -> [*,*,*,*]	675	3.7022817
[*,3,*,*] -> [*,4,*,*]	222	1.2176393
[*,3,*,4] -> [*,*,*,4]	207	1.1353664
[*,3,4,*] -> [*,*,4,*]	223	1.2231241
[*,4,*,*] -> [*,*,*,*]	1598	8.764809
[*,4,*,*] -> [*,2,*,*]	227	1.2450637
[*,4,*,*] -> [*,3,*,*]	296	1.6235191
[*,4,*,*] -> [*,4,*,*]	2395	13.136244
[*,4,*,*] -> [*,4,*,1]	342	1.8758228
[*,4,*,*] -> [*,4,*,9]	191	1.0476086
[*,4,*,*] -> [*,4,1,*]	321	1.7606406
[*,4,*,*] -> [*,4,3,*]	187	1.0256691
[*,4,*,*] -> [*,4,9,*]	213	1.1682756
[*,4,*,*] -> [*,6,*,*]	302	1.6564282
[*,4,*,*] -> [*,9,*,*]	277	1.5193067
[*,4,*,*] -> [1,4,*,*]	434	2.38043
[*,4,*,*] -> [3,4,*,*]	188	1.031154
[*,4,*,*] -> [9,4,*,*]	236	1.2944274
[*,4,*,1] -> [*,4,*,*]	292	1.6015797
[*,4,*,3] -> [*,4,*,*]	185	1.0146995
[*,4,1,*] -> [*,4,*,*]	312	1.7112769
[*,4,3,*] -> [*,4,*,*]	254	1.3931549
[*,5,*,*] -> [*,*,*,*]	308	1.6893374
[*,6,*,*] -> [*,*,*,*]	576	3.1592803
[*,6,*,*] -> [*,4,*,*]	307	1.6838526
[*,6,*,*] -> [*,6,*,*]	362	1.98552
[*,9,*,*] -> [*,*,*,*]	1058	5.8029838
[*,9,*,*] -> [*,4,*,*]	424	2.3255813
[*,9,*,*] -> [*,9,*,*]	245	1.3437911
[1,*,*,*] -> [*,*,*,*]	412	2.259763
[1,*,*,4] -> [*,*,*,4]	394	2.1610355
[1,*,4,*] -> [*,*,4,*]	413	2.2652478
[1,4,*,*] -> [*,4,*,*]	408	2.2378237
[11,*,*,*] -> [*,*,*,*]	327	1.7935498
[16,*,*,*] -> [*,*,*,*]	271	1.4863975
[2,*,*,*] -> [*,*,*,*]	531	2.9124615
[2,*,*,*] -> [3,*,*,*]	188	1.031154
[2,*,*,*] -> [4,*,*,*]	681	3.7351909
[2,*,*,*] -> [6,*,*,*]	314	1.7222466
[3,*,*,*] -> [*,*,*,*]	765	4.195919
[3,*,*,*] -> [4,*,*,*]	350	1.9197016
[3,*,*,4] -> [*,*,*,4]	222	1.2176393
[3,*,4,*] -> [*,*,4,*]	258	1.4150944
[3,4,*,*] -> [*,4,*,*]	258	1.4150944
[4,*,*,*] -> [*,*,*,*]	2246	12.318999
[4,*,*,*] -> [2,*,*,*]	413	2.2652478
[4,*,*,*] -> [3,*,*,*]	363	1.9910048
[4,*,*,*] -> [4,*,*,*]	2996	16.432646
[4,*,*,*] -> [4,*,*,1]	430	2.3584905
[4,*,*,*] -> [4,*,*,3]	250	1.3712155
[4,*,*,*] -> [4,*,*,9]	254	1.3931549
[4,*,*,*] -> [4,*,1,*]	426	2.3365512
[4,*,*,*] -> [4,*,3,*]	215	1.1792452
[4,*,*,*] -> [4,*,9,*]	246	1.349276
[4,*,*,*] -> [4,1,*,*]	456	2.501097
[4,*,*,*] -> [4,3,*,*]	229	1.2560333
[4,*,*,*] -> [4,9,*,*]	289	1.5851251
[4,*,*,*] -> [6,*,*,*]	435	2.3859148
[4,*,*,*] -> [9,*,*,*]	319	1.7496709
[4,*,*,1] -> [4,*,*,*]	414	2.2707329
[4,*,*,3] -> [4,*,*,*]	297	1.629004
[4,*,1,*] -> [4,*,*,*]	397	2.1774902
[4,*,3,*] -> [4,*,*,*]	275	1.508337
[4,1,*,*] -> [4,*,*,*]	440	2.4133391
[4,3,*,*] -> [4,*,*,*]	319	1.7496709
[4,9,*,*] -> [4,*,*,*]	186	1.0201843
[5,*,*,*] -> [*,*,*,*]	537	2.9453707
[6,*,*,*] -> [*,*,*,*]	857	4.7005267
[6,*,*,*] -> [4,*,*,*]	391	2.1445808
[6,*,*,*] -> [6,*,*,*]	407	2.2323387
[9,*,*,*] -> [*,*,*,*]	1249	6.850592
[9,*,*,*] -> [4,*,*,*]	524	2.8740675
[9,*,*,*] -> [6,*,*,*]	211	1.1573058
[9,*,*,*] -> [9,*,*,*]	337	1.8483984
[9,*,*,*] -> [9,*,*,9]	234	1.2834576
[9,*,*,*] -> [9,*,9,*]	184	1.0092145
[9,*,*,9] -> [9,*,*,*]	229	1.2560333
[9,9,*,*] -> [9,*,*,*]	217	1.190215

Appendix B1 Experiment 1.1

Feature sets: DA + last word

Accepted / Bad: 14282 / 5689

Goups: backchannel (Ba),non-backchannel (NBa)

Classifier: J48 decision tree (default arguments)

Traning / testing: 10 fold cross validation

Dataset	Correct classifications	# Instances	Precision	Recall	f-Measure
All combined Instances: 14282	61.7 %	Ba: 7114 NBa: 7168	Ba: 0.61 NBa: 0.624	Ba: 0.624 NBa: 0.609	Ba: 0.612 NBa: 0.612
ES2002 ~ ES2016 Instances: 6079	61.1 %	Ba: 3036 NBa: 3043	Ba: 0.602 NBa: 0.623	Ba: 0.654 NBa: 0.569	Ba: 0.627 NBa: 0.594
IS1000 ~ IS1009 Instances: 4241	60.8 %	Ba: 2125 NBa: 2116	Ba: 0.602 NBa: 0.615	Ba: 0.644 NBa: 0.572	Ba: 0.601 NBa: 0.601
TS3003 ~ TS3012 Instances: 3962	62.1 %	Ba: 1953 NBa: 2009	Ba: 0.61 NBa: 0.632	Ba: 0.624 NBa: 0.618	Ba: 0.621 NBa: 0.621

Feature merit ranking

Dataset: All: ES2002 ~ TS3012

Evaluator: InfoGainAttributeEval (default arguments)

Method: 10 fold cross validation (list only features with average merit > 0)

average merit	average rank	attribute	average merit	average rank	attribute
0.037 +- 0.001	1 +- 0	2 word_amnt	0.001 +- 0	13 +- 0.63	10 w_p_m_dif
0.034 +- 0.001	2 +- 0	3 da_dur	0 +- 0	14.2 +- 0.4	9 w_i_m_dif
0.028 +- 0.001	3 +- 0	1 dag_label	0 +- 0	15.2 +- 0.4	11 w_p_vf_dif
0.015 +- 0.001	4 +- 0	16 w_pau_dur	0 +- 0.001	15.4 +- 1.2	6 da_i_mean
0.012 +- 0	5.2 +- 0.4	4 da_w_l_avg			
0.011 +- 0.001	6 +- 0.63	7 da_vf_r			
0.01 +- 0.001	6.8 +- 0.4	8 da_vf_ps			
0.006 +- 0	8 +- 0	15 w_avg_l_dif			
0.004 +- 0	9 +- 0	5 da_p_mean			
0.002 +- 0	10.2 +- 0.4	12 w_i_RFC			
0.002 +- 0	10.8 +- 0.4	13 w_p_RFC			
0.001 +- 0	12.2 +- 0.4	14 w_dur			

Appendix B2 Experiment 1.2

Feature sets: DA + last 300 ms

Goups: backchannel (Ba) non-backchannel (NBa)

Classifier: J48 decision tree (default arguments)

Traning / testing: 10 fold cross validation

Dataset	Correct classifications	# Instances	Precision	Recall	f-Measure
All combined Instances: 14282	61.8%	Ba: 7114 NBa: 7168	Ba: 0.61 NBa: 0.627	Ba: 0.628 NBa: 0.607	Ba: 0.632 NBa: 0.632
ES2002 ~ ES2016 Instances: 6079	61.7 %	Ba: 3036 NBa: 3043	Ba: 0.608 NBa: 0.627	Ba: 0.655 NBa: 0.579	Ba: 0.631 NBa: 0.602
IS1000 ~ IS1009 Instances: 4241	60.2 %	Ba: 2125 NBa: 2116	Ba: 0.598 NBa: 0.606	Ba: 0.628 NBa: 0.575	Ba: 0.613 NBa: 0.59
TS3003 ~ TS3012 Instances: 3962	60.4 %	Ba: 1953 NBa: 2009	Ba: 0.587 NBa: 0.626	Ba: 0.666 NBa: 0.544	Ba: 0.624 NBa: 0.582

Feature merit ranking

Dataset: All: ES2002 ~ TS3012

Evaluator: InfoGainAttributeEval (default arguments)

Method: 10 fold cross validation (list only features with average merit > 0)

average merit	average rank	attribute	average merit	average rank	attribute
0.037 +- 0.001	1 +- 0	2 word_amnt			
0.034 +- 0.001	2 +- 0	3 da_dur			
0.028 +- 0.001	3 +- 0	1 dag_label			
0.012 +- 0	4.2 +- 0.4	4 da_w_l_avg			
0.011 +- 0.001	5 +- 0.63	7 da_vf_r			
0.01 +- 0.001	5.8 +- 0.4	8 da_vf_ps			
0.006 +- 0	7 +- 0	12 300_i_RFC			
0.005 +- 0	8.1 +- 0.3	13 300_p_RFC			
0.004 +- 0	9.1 +- 0.54	11 300_p_vf_dif			
0.004 +- 0	9.8 +- 0.4	5 da_p_mean			
0.003 +- 0	11 +- 0	9 300_i_m_dif			
0.002 +- 0	12 +- 0	10 300_p_m_dif			
0 +- 0.001	13 +- 0	6 da_i_mean			

Appendix B3 Experiment 1.3

Feature sets: DA + last 500 ms

Goups: backchannel (Ba) non-backchannel (NBa)

Classifier: J48 decision tree (default arguments)

Traning / testing: 10 fold cross validation

Dataset	Correct classifications	# Instances	Precision	Recall	f-Measure
All combined Instances: 14282	61.8 %	Ba: 7114 NBa: 7168	Ba: 0.61 NBa: 0.627	BA: 0.646 NBa: 0.589	Ba: 0.628 NBa: 0.607
ES2002 ~ ES2016 Instances: 6079	61.7 %	Ba: 3036 NBa: 3043	Ba: 0.608 NBa: 0.627	Ba: 0.655 NBa: 0.579	Ba: 0.631 NBa: 0.602
IS1000 ~ IS1009 Instances: 4241	60.2%	Ba: 2125 NBa: 2116	Ba: 0.598 NBa: 0.606	Ba: 0.628 NBa: 0.575	Ba: 0.613 NBa: 0.59
TS3003 ~ TS3012 Instances: 3962	60.2 %	Ba: 1953 NBa: 2009	Ba: 0.589 NBa: 0.622	Ba: 0.657 NBa: 0.548	Ba: 0.619 NBa: 0.583

Feature merit ranking

Dataset: All: ES2002 ~ TS3012

Evaluator: InfoGainAttributeEval (default arguments)

Method: 10 fold cross validation (list only features with average merit > 0)

average merit	average rank	attribute	average merit	average rank	attribute
0.037 +- 0.001	1 +- 0	2 word_amnt			
0.034 +- 0.001	2 +- 0	3 da_dur			
0.028 +- 0.001	3 +- 0	1 dag_label			
0.012 +- 0	4.2 +- 0.4	4 da_w_l_avg			
0.011 +- 0.001	5 +- 0.63	7 da_vf_r			
0.01 +- 0.001	5.8 +- 0.4	8 da_vf_ps			
0.006 +- 0	7 +- 0	12 500_i_RFC			
0.005 +- 0	8.1 +- 0.3	13 500_p_RFC			
0.004 +- 0	9.1 +- 0.54	11 500_p_vf_dif			
0.004 +- 0	9.8 +- 0.4	5 da_p_mean			
0.003 +- 0	11 +- 0	9 500_i_m_dif			
0.002 +- 0	12 +- 0	10 500_p_m_dif			
0 +- 0.001	13 +- 0	6 da_i_mean			

Appendix B4 Experiment 1.4

Feature sets: DA + last1000 ms

Goups: backchannel (Ba) non-backchannel (NBa)

Classifier: J48 decision tree (default arguments)

Traning / testing: 10 fold cross validation

Dataset	Correct classifications	# Instances	Precision	Recall	f-Measure
All combined Instances: 14282	61.8 %	Ba: 7114 NBa: 7168	Ba: 0.61 NBa: 0.627	BA: 0.646 NBa: 0.589	Ba: 0.628 NBa: 0.607
ES2002 ~ ES2016 Instances: 6079	61.7 %	Ba: 3036 NBa: 3043	Ba: 0.608 NBa: 0.627	Ba: 0.655 NBa: 0.602	Ba: 0.622 NBa: 0.622
IS1000 ~ IS1009 Instances: 4241	60.2 %	Ba: 2125 NBa: 2116	Ba: 0.598 NBa: 0.606	Ba: 0.628 NBa: 0.575	Ba: 0.613 NBa: 0.59
TS3003 ~ TS3012 Instances: 3962	60.4%	Ba: 1953 NBa: 2009	Ba: 0.5887 NBa: 0.626	Ba: 0.666 NBa: 0.544	Ba: 0.624 NBa: 0.582

Feature merit ranking

Dataset: All: ES2002 ~ TS3012

Evaluator: InfoGainAttributeEval (default arguments)

Method: 10 fold cross validation (list only features with average merit > 0)

average merit	average rank	attribute	average merit	average rank	attribute
0.03728	2	word_amnt			
0.03363	3	da_dur			
0.02777	1	dag_label			
0.01218	4	da_w_l_avg			
0.01144	7	da_vf_r			
0.01092	8	da_vf_ps			
0.00557	12	s_i_RFC			
0.00468	13	s_p_RFC			
0.00408	11	s_p_vf_dif			
0.00381	5	da_p_mean			
0.00304	9	s_i_m_dif			
0.00187	10	s_p_m_dif			
0	6	da_i_mean			

Appendix B5 Experiment 1.5

Feature sets: DA +last word+ last 300ms + last 500 ms + last 1000 ms

Goups: backchannel (Ba) non-backchannel (NBa)

Classifier: J48 decision tree (default arguments)

Traning / testing: 10 fold cross validation

Dataset	Correct classifications	# Instances	Precision	Recall	f-Measure
All combined Instances: 14282	61.9 %	Ba: 7114 NBa: 7168	Ba: 0.607 NBa: 0.632	BA: 0.664 NBa: 0.574	Ba: 0.634 NBa: 0.602
ES2002 ~ ES2016 Instances: 6079	61.9 %	Ba: 3036 NBa: 3043	Ba: 0.616 NBa: 0.622	Ba: 0.628 NBa: 0.61	Ba: 0.622 NBa: 0.616
IS1000 ~ IS1009 Instances: 4241	60.4 %	Ba: 2125 NBa: 2116	Ba: 0.605 NBa: 0.605	Ba: 0.604 NBa: 0.604	Ba: 0.604 NBa: 0.603
TS3003 ~ TS3012 Instances: 3962	60.8 %	Ba: 1953 NBa: 2009	Ba: 0.593 NBa: 0.615	Ba: 0.624 NBa: 0.584	Ba: 0.608 NBa: 0.599

Feature merit ranking

Dataset: All: ES2002 ~ TS3012

Evaluator: InfoGainAttributeEval (default arguments)

Method: 10 fold cross validation (list only features with average merit > 0)

average merit	average rank	attribute	average merit	average rank	attribute
0.037 +- 0.001	1 +- 0	2 word_amnt	0.004 +- 0	16 +- 1	19 s_p_vf_dif
0.034 +- 0.001	2 +- 0	3 da_dur	0.004 +- 0	16.1 +- 1.3	24 500_p_vf_dif
0.028 +- 0.001	3 +- 0	1 dag_label	0.004 +- 0	17.4 +- 1.2	5 da_p_mean
0.015 +- 0.001	4 +- 0	16 w_pau_dur	0.003 +- 0	19.3 +- 0.64	27 300_i_m_dif
0.012 +- 0	5.2 +- 0.4	4 da_w_l_avg	0.003 +- 0	20.3 +- 0.46	22 500_i_m_dif
0.011 +- 0.001	6 +- 0.63	7 da_vf_r	0.003 +- 0	20.4 +- 0.8	17 s_i_m_dif
0.01 +- 0.001	6.8 +- 0.4	8 da_vf_ps	0.002 +- 0	23.5 +- 0.5	23 500_p_m_dif
0.006 +- 0	8.6 +- 1.2	15 w_avg_l_dif	0.002 +- 0	23.6 +- 1.69	18 s_p_m_dif
0.006 +- 0	8.9 +- 0.54	20 s_i_RFC	0.002 +- 0	23.7 +- 1.55	12 w_i_RFC
0.006 +- 0	10.1 +- 0.54	25 500_i_RFC	0.002 +- 0	24.3 +- 0.9	28 300_p_m_dif
0.006 +- 0	10.4 +- 0.8	30 300_i_RFC	0.002 +- 0	24.9 +- 1.51	13 w_p_RFC
0.005 +- 0	12.8 +- 1.54	26 500_p_RFC	0.001 +- 0	27.2 +- 0.4	14 w_dur
0.005 +- 0	13.5 +- 0.92	31 300_p_RFC	0.001 +- 0	28 +- 0.63	10 w_p_m_dif
0.005 +- 0	13.6 +- 0.92	21 s_p_RFC	0 +- 0	29.2 +- 0.4	9 w_i_m_dif
0.004 +- 0	15.6 +- 1.56	29 300_p_vf_dif	0 +- 0	30.2 +- 0.4	11 w_p_vf_dif
			0 +- 0.001	30.4 +- 1.2	6 da_i_mean

AppendixC1 Experiment 2.1

Feature sets: DA + last word

Accepted / Bad:

Goups: backchannel (Ba),non-backchannel (NBa)

Classifier: J48 decision tree (default arguments)

Traning / testing: 10 fold cross validation

Dataset	Correct classifications	# Instances	Precision	Recall	f-Measure
All combined Instances: 18424	65.1%	Ba: 8761 NBa: 9663	Ba: 0.635 NBa: 0.666	BA: 0.628 NBa: 0.672	Ba: 0.632 NBa: 0.669
ES2002 ~ ES2016 Instances: 7936	67.2 %	Ba: 3767 NBa: 4169	Ba: 0.651 NBa: 0.691	Ba: 0.665 NBa: 0.678	Ba: 0.658 NBa: 0.685
IS1000 ~ IS1009 Instances: 5331	64.0 %	Ba: 2541 NBa: 2790	Ba: 0.61 NBa: 0.673	Ba: 0.677 NBa: 0.606	Ba: 0.642 NBa: 0.638
TS3003 ~ TS3012 Instances: 5157	63.9%	Ba: 2453 NBa: 2704	Ba: 0.628 NBa: 0.647	Ba: 0.59 NBa: 0.683	Ba: 0.609 NBa: 0.665

Feature merit ranking

Dataset: All: ES2002 ~ TS3012

Evaluator: InfoGainAttributeEval (default arguments)

Method: 10 fold cross validation (list only features with average merit > 0)

average merit	average rank	attribute
0.083 +- 0.001	1 +- 0	2 word_amnt
0.081 +- 0.001	2 +- 0	7 da_vfrms
0.074 +- 0.001	3 +- 0	3 da_dur
0.034 +- 0.001	4 +- 0	1 dag_label
0.029 +- 0.001	5 +- 0	19 w_pau_dur
0.028 +- 0.001	6 +- 0	4 da_w_l_avg
0.026 +- 0.001	7 +- 0	8 da_vf_r
0.022 +- 0.001	8 +- 0	9 da_vf_ps
0.02 +- 0.001	9 +- 0	11 da_i_sd
0.012 +- 0.001	10 +- 0	18 w_avg_l_dif
0.01 +- 0.001	11.2 +- 0.6	17 w_dur
0.008 +- 0.001	12.3 +- 0.46	6 da_p_sd
0.008 +- 0.001	12.5 +- 0.67	5 da_p_mean
0.005 +- 0	14 +- 0	13 w_p_m_dif
0.003 +- 0	15.4 +- 0.49	10 da_i_mean
0.003 +- 0	15.7 +- 0.64	15 w_i_RFC
0.002 +- 0	16.9 +- 0.3	16 w_p_RFC
0.001 +- 0	18 +- 0	12 w_i_m_dif
0 +- 0	19 +- 0	14 w_p_vf_dif

Appendix C2 Experiment 2.2

Feature sets: DA + last 300 ms

Goups: backchannel (Ba) non-backchannel (NBa)

Classifier: J48 decision tree (default arguments)

Traning / testing: 10 fold cross validation

Dataset	Correct classifications	# Instances	Precision	Recall	f-Measure
All combined Instances: 18424	65.4%	Ba: 8761 NBa: 9663	Ba: 0.634 NBa: 0.674	BA: 0.648 NBa: 0.66	Ba: 0.641 NBa: 0.667
ES2002 ~ ES2016 Instances: 7936	67.7%	Ba: 3767 NBa: 4169	Ba: 0.655 NBa: 0.699	Ba: 0.677 NBa: 0.677	Ba: 0.666 NBa: 0.668
IS1000 ~ IS1009 Instances: 5331	64.2 %	Ba: 2541 NBa: 2790	Ba: 0.628 NBa: 0.654	Ba: 0.61 NBa: 0.671	Ba: 0.619 NBa: 0.662
TS3003 ~ TS3012 Instances: 5157	63.9%	Ba: 2453 NBa: 2704	Ba: 0.633 NBa: 0.643	Ba: 0.573 NBa: 0.699	Ba: 0.602 NBa: 0.67

Feature merit ranking

Dataset: All: ES2002 ~ TS3012

Evaluator: InfoGainAttributeEval (default arguments)

Method: 10 fold cross validation (list only features with average merit > 0)

average merit	average rank	attribute	average merit	average rank	attribute
0.083 +- 0.001	1 +- 0	2 word_amnt	0.008 +- 0.001	9.4 +- 0.49	6 da_p_sd
0.081 +- 0.001	2 +- 0	7 da_vfrms	0.008 +- 0.001	9.6 +- 0.49	5 da_p_mean
0.074 +- 0.001	3 +- 0	3 da_dur	0.005 +- 0	11.4 +- 0.49	14 300_p_vf_dif
0.034 +- 0.001	4 +- 0	1 dag_label	0.005 +- 0	11.9 +- 0.7	13 300_p_m_dif
0.028 +- 0.001	5 +- 0	4 da_w_l_avg	0.005 +- 0	12.8 +- 0.75	12 300_i_m_dif
0.026 +- 0.001	6 +- 0	8 da_vf_r	0.004 +- 0	13.9 +- 0.3	16 300_p_RFC
0.022 +- 0.001	7 +- 0	9 da_vf_ps	0.003 +- 0	15 +- 0	10 da_i_mean
0.02 +- 0.001	8 +- 0	11 da_i_sd	0.002 +- 0	16 +- 0	15 300_i_RFC

Appendix C3 Experiment 2.3

Feature sets: DA + last 500 ms

Goups: backchannel (Ba) non-backchannel (NBa)

Classifier: J48 decision tree (default arguments)

Traning / testing: 10 fold cross validation

Dataset	Correct classifications	# Instances	Precision	Recall	f-Measure
All combined Instances: 18424	65.5%	Ba: 8761 NBa: 9663	Ba: 0.638 NBa: 0.669	BA: 0.631 NBa: 0.676	Ba: 0.635 NBa: 0.672
ES2002 ~ ES2016 Instances: 7936	67.8%	Ba: 3767 NBa: 4169	Ba: 0.656 NBa: 0.699	Ba: 0.677 NBa: 0.68	Ba: 0.666 NBa: 0.689
IS1000 ~ IS1009 Instances: 5331	63.6 %	Ba: 2541 NBa: 2790	Ba: 0.623 NBa: 0.648	Ba: 0.601 NBa: 0.669	Ba: 0.612 NBa: 0.658
TS3003 ~ TS3012 Instances: 5157	63.2%	Ba: 2453 NBa: 2704	Ba: 0.624 NBa: 0.638	Ba: 0.57 NBa: 0.688	Ba: 0.596 NBa: 0.662

Feature merit ranking

Dataset: All: ES2002 ~ TS3012

Evaluator: InfoGainAttributeEval (default arguments)

Method: 10 fold cross validation (list only features with average merit > 0)

average merit	average rank	attribute	average merit	average rank	attribute
0.083 +- 0.001	1 +- 0	2 word_amnt	0.008 +- 0.001	9.4 +- 0.49	6 da_p_sd
0.081 +- 0.001	2 +- 0	7 da_vfrms	0.008 +- 0.001	9.6 +- 0.49	5 da_p_mean
0.074 +- 0.001	3 +- 0	3 da_dur	0.005 +- 0	11.4 +- 0.49	15 500_i_RFC
0.034 +- 0.001	4 +- 0	1 dag_label	0.005 +- 0	11.6 +- 0.49	13 500_p_m_dif
0.028 +- 0.001	5 +- 0	4 da_w_l_avg	0.004 +- 0	13 +- 0	16 500_p_RFC
0.026 +- 0.001	6 +- 0	8 da_vf_r	0.003 +- 0	14 +- 0	10 da_i_mean
0.022 +- 0.001	7 +- 0	9 da_vf_ps	0.003 +- 0	15 +- 0	14 500_p_vf_dif
0.02 +- 0.001	8 +- 0	11 da_i_sd	0.002 +- 0	16 +- 0	12 500_i_m_dif

Appendix C4 Experiment 2.4

Feature sets: DA + last 1000 ms

Goups: backchannel (Ba) non-backchannel (NBa)

Classifier: J48 decision tree (default arguments)

Traning / testing: 10 fold cross validation

Dataset	Correct classifications	# Instances	Precision	Recall	f-Measure
All combined Instances: 18424	65.9%	Ba: 8761 NBa: 9663	Ba: 0.637 NBa: 0.681	BA: 0.659 NBa: 0.659	Ba: 0.648 NBa: 0.67
ES2002 ~ ES2016 Instances: 7936	67.4%	Ba: 3767 NBa: 4169	Ba: 0.646 NBa: 0.702	Ba: 0.691 NBa: 0.659	Ba: 0.668 NBa: 0.68
IS1000 ~ IS1009 Instances: 5331	63.9 %	Ba: 2541 NBa: 2790	Ba: 0.62 NBa: 0.656	Ba: 0.625 NBa: 0.651	Ba: 0.623 NBa: 0.654
TS3003 ~ TS3012 Instances: 5157	64.4%	Ba: 2453 NBa: 2704	Ba: 0.633 NBa: 0.653	Ba: 0.598 NBa: 0.685	Ba: 0.615 NBa: 0.669

Feature merit ranking

Dataset: All: ES2002 ~ TS3012

Evaluator: InfoGainAttributeEval (default arguments)

Method: 10 fold cross validation (list only features with average merit > 0)

```
average merit   average rank    attribute
0.083 +- 0.001    1   +- 0       2
word_amnt
 0.081 +- 0.001    2   +- 0       7 da_vfrms
 0.074 +- 0.001    3   +- 0       3 da_dur
 0.034 +- 0.001    4   +- 0       1 dag_label
 0.028 +- 0.001    5   +- 0       4
da_w_l_avg
 0.026 +- 0.001    6   +- 0       8 da_vf_r
 0.022 +- 0.001    7   +- 0       9 da_vf_ps
 0.02  +- 0.001    8   +- 0     11 da_i_sd
```

```
average merit   average rank    attribute
0.017 +- 0.001     9   +- 0      15 1s_i_RFC
 0.008 +- 0       10.4 +- 0.66   13
1s_p_m_dif
 0.008 +- 0.001    11.1 +- 0.7     6 da_p_sd
 0.008 +- 0.001    11.5 +- 0.67    5
da_p_mean
 0.003 +- 0       13.2 +- 0.4    10
da_i_mean
 0.003 +- 0       13.8 +- 0.4    16 1s_p_RFC
 0.001 +- 0       15   +- 0      12 1s_i_m_dif
0     +- 0       16   +- 0      14 1s_p_vf_dif
```

Appendix C5 Experiment 2.5

Feature sets: DA +last word+ last 300ms + last 500 ms + last 1000 ms

Goups: backchannel (Ba) non-backchannel (NBa)

Classifier: J48 decision tree (default arguments)

Traning / testing: 10 fold cross validation

Dataset	Correct classifications	# Instances	Precision	Recall	f-Measure
All combined Instances: 18424	65.4 %	Ba: 8761 NBa: 9663	Ba: 0.633 NBa: 0.674	BA: 0.648 NBa: 0.659	Ba: 0.64 NBa: 0.666
ES2002 ~ ES2016 Instances: 7936	67.3 %	Ba: 3767 NBa: 4169	Ba: 0.653 NBa: 0.692	Ba: 0.665 NBa: 0.68	Ba: 0.659 NBa: 0.686
IS1000 ~ IS1009 Instances: 5331	63.1%	Ba: 2541 NBa: 2790	Ba: 0.604 NBa: 0.659	Ba: 0.654 NBa: 0.61	Ba: 0.628 NBa: 0.634
TS3003 ~ TS3012 Instances: 5157	63.9%	Ba: 2453 NBa: 2704	Ba: 0.629 NBa: 0.648	Ba: 0.591 NBa: 0.683	Ba: 0.609 NBa: 0.665

Feature merit ranking

Dataset: All: ES2002 ~ TS3012

Evaluator: InfoGainAttributeEval (default arguments)

Method: 10 fold cross validation (list only features with average merit > 0)

average merit	average rank	attribute	average merit	average rank	attribute
0.083 +- 0.001	1 +- 0	2 word_amnt	0.005 +- 0	17.7 +- 1.19	32 300_p_vf_dif
0.081 +- 0.001	2 +- 0	7 da_vfrms	0.005 +- 0	18.7 +- 1	31 300_p_m_dif
0.074 +- 0.001	3 +- 0	3 da_dur	0.005 +- 0	20.1 +- 1.04	30 300_i_m_dif
0.034 +- 0.001	4 +- 0	1 dag_label	0.005 +- 0	20.6 +- 0.8	13 w_p_m_dif
0.029 +- 0.001	5 +- 0	19 w_pau_dur	0.004 +- 0	21.8 +- 0.6	34 300_p_RFC
0.028 +- 0.001	6 +- 0	4 da_w_l_avg	0.004 +- 0	23 +- 0	29 500_p_RFC
0.026 +- 0.001	7 +- 0	8 da_vf_r	0.003 +- 0	24.6 +- 0.8	10 da_i_mean
0.022 +- 0.001	8 +- 0	9 da_vf_ps	0.003 +- 0	25.2 +- 0.98	15 w_i_RFC
0.02 +- 0.001	9 +- 0	11 da_i_sd	0.003 +- 0	26 +- 1.41	24 1s_p_RFC
0.017 +- 0.001	10 +- 0	23 1s_i_RFC	0.003 +- 0	27.1 +- 0.83	27 500_p_vf_dif
0.012 +- 0.001	11 +- 0	18 w_avg_l_dif	0.002 +- 0	27.7 +- 1.19	16 w_p_RFC
0.01 +- 0.001	12.2 +- 0.6	17 w_dur	0.002 +- 0	28.5 +- 1.02	33 300_i_RFC
0.008 +- 0	13.4 +- 0.66	21 1s_p_m_dif	0.002 +- 0	29.9 +- 0.3	25 500_i_m_dif
0.008 +- 0.001	14 +- 0.77	6 da_p_sd	0.001 +- 0	31 +- 0	12 w_i_m_dif
0.008 +- 0.001	14.4 +- 0.92	5 da_p_mean	0.001 +- 0	32 +- 0	20 1s_i_m_dif
0.005 +- 0	16.8 +- 0.87	28 500_i_RFC	0 +- 0	33 +- 0	14 w_p_vf_dif
0.005 +- 0	17.3 +- 1.27	26 500_p_m_dif	0 +- 0	34 +- 0	22 1s_p_vf_dif

Appendix D1 Experiment 3.1

Feature sets: Turn + 200ms +400ms +600ms +800ms + 1000ms

Goups: backchannel (Ba) non-backchannel (NBa)

Classifier: J48 decision tree (default arguments)

Traning / testing: 10 fold cross validation

Dataset	Correct classifications	# Instances	Precision	Recall	f-Measure
All combined Instances: 16987	62.9%	Ba: 8461 NBa: 8526	Ba: 0.624 NBa: 0.634	BA: 0.633 NBa: 0.626	Ba: 0.633 NBa: 0.626

Feature merit ranking

Dataset: All: ES2002 ~ TS3012

Evaluator: InfoGainAttributeEval (default arguments)

Method: 10 fold cross validation (list only features with average merit > 0)

average merit	average rank	attribute
0.07 +- 0.001	1.1 +- 0.3	2 word_amnt
0.069 +- 0.001	1.9 +- 0.3	7 da_vfrms
0.058 +- 0.001	3 +- 0	3 da_dur
0.028 +- 0.001	4 +- 0	1 dag_label
0.018 +- 0.001	5 +- 0	8 da_vf_r
0.015 +- 0	6 +- 0	9 da_vf_ps
0.014 +- 0.001	7 +- 0	11 da_i_sd
0.011 +- 0.001	8 +- 0	4 da_w_l_avg
0.008 +- 0	9 +- 0	12 it1000_p_mean_d
0.006 +- 0	10 +- 0	15 it1000_i_mean_d
0.004 +- 0.001	12 +- 1.55	18 it800_p_sd_d
0.004 +- 0	12.5 +- 0.81	13 it1000_p_sd_d
0.004 +- 0	12.8 +- 1.08	5 da_p_mean
0.004 +- 0.001	13.4 +- 1.62	6 da_p_sd
0.003 +- 0	14.5 +- 0.67	19 it800_vfr_d
0.003 +- 0	17 +- 1.18	17 it800_p_mean_d
0.003 +- 0	17.1 +- 1.04	21 it800_i_sd_d
0.003 +- 0	18 +- 0.89	14 it1000_vfr_d
0.003 +- 0	18.1 +- 1.7	16 it1000_i_sd_d
0.002 +- 0.001	19.9 +- 0.94	23 it600_p_sd_d
0.001 +- 0	21.7 +- 0.64	24 it600_vfr_d
0.001 +- 0	23.2 +- 4.17	25 it600_i_mean_d
0 +- 0	25.2 +- 3.03	32 it200_p_mean_d
0 +- 0	26.2 +- 1.08	36 it200_i_sd_d
0 +- 0	26.9 +- 0.83	35 it200_i_mean_d
0 +- 0	27.5 +- 1.02	33 it200_p_sd_d
0 +- 0	27.7 +- 1.49	34 it200_vfr_d
0 +- 0.001	28.2 +- 3.87	22 it600_p_mean_d
0 +- 0.001	28.5 +- 4.72	27 it400_p_mean_d
0.001 +- 0.001	28.9 +- 6.66	10 da_i_mean
0 +- 0	29.9 +- 4.87	31 it400_i_sd_d
0 +- 0	30.5 +- 1.28	26 it600_i_sd_d
0 +- 0	31.4 +- 0.49	20 it800_i_mean_d
0 +- 0	31.7 +- 3.95	30 it400_i_mean_d
0 +- 0	33.2 +- 2.27	28 it400_p_sd_d
0 +- 0	35 +- 1	29 it400_vfr_d

Appendix D2 Experiment 3.2

Feature sets: Turn + 200ms +400ms +600ms + 800ms

Goups: backchannel (Ba) non-backchannel (NBa)

Classifier: J48 decision tree (default arguments)

Traning / testing: 10 fold cross validation

Dataset	Correct classifications	# Instances	Precision	Recall	f-Measure
All combined Instances:	63.3%	Ba: 8461 NBa: 8526	Ba: 0.626 NBa: 0.639	BA: 0.649 NBa: 0.615	Ba: 0.638 NBa: 0.627

Feature merit ranking

Dataset: All: ES2002 ~ TS3012

Evaluator: InfoGainAttributeEval (default arguments)

Method: 10 fold cross validation (list only features with average merit > 0)

average merit	average rank	attribute
0.07 +- 0.001	1.1 +- 0.3	2 word_amnt
0.069 +- 0.001	1.9 +- 0.3	7 da_vfrms
0.058 +- 0.001	3 +- 0	3 da_dur
0.028 +- 0.001	4 +- 0	1 dag_label
0.018 +- 0.001	5 +- 0	8 da_vf_r
0.015 +- 0	6 +- 0	9 da_vf_ps
0.014 +- 0.001	7 +- 0	11 da_i_sd
0.011 +- 0.001	8 +- 0	4 da_w_l_avg
0.004 +- 0.001	9.7 +- 1.19	13 it800_p_sd_d
0.004 +- 0	10.1 +- 0.7	5 da_p_mean
0.004 +- 0.001	10.8 +- 1.08	6 da_p_sd
0.003 +- 0	11.5 +- 0.67	14 it800_vfr_d
0.003 +- 0	13.4 +- 0.66	12 it800_p_mean_d
0.003 +- 0	13.5 +- 0.5	16 it800_i_sd_d
0.002 +- 0.001	15.3 +- 0.46	18 it600_p_sd_d
0.001 +- 0	16.7 +- 0.64	19 it600_vfr_d
0.001 +- 0	17.7 +- 2.79	20 it600_i_mean_d
0 +- 0	19.9 +- 1.64	27 it200_p_mean_d
0 +- 0	20.9 +- 2.77	26 it400_i_sd_d
0 +- 0	21.6 +- 1.43	28 it200_p_sd_d
0 +- 0	22.6 +- 0.92	31 it200_i_sd_d
0 +- 0	23 +- 1.73	30 it200_i_mean_d
0 +- 0.001	23.1 +- 3.78	17 it600_p_mean_d
0 +- 0	23.8 +- 0.4	29 it200_vfr_d
0.001 +- 0.001	24 +- 6.71	10 da_i_mean
0 +- 0.001	24.9 +- 5.54	22 it400_p_mean_d
0 +- 0	25.3 +- 4.98	25 it400_i_mean_d
0 +- 0	26.7 +- 1.68	21 it600_i_sd_d
0 +- 0	27 +- 0.45	15 it800_i_mean_d
0 +- 0	29 +- 1.55	23 it400_p_sd_d
0 +- 0	29.5 +- 1.12	24 it400_vfr_d

Appendix D3 Experiment 3.3

Feature sets:	Turn + 200ms +400ms +600ms
Goups:	backchannel (Ba) non-backchannel (NBa)
Classifier:	J48 decision tree (default arguments)
Traning / testing:	10 fold cross validation

Dataset	Correct classifications	# Instances	Precision	Recall	f-Measure
All combined Instances:	63.4%	Ba: 8461 NBa: 8526	Ba: 0.625 NBa: 0.643	BA: 0.66 NBa: 0.608	Ba: 0.642 NBa: 0.625

Feature merit ranking

Dataset:	All: ES2002 ~ TS3012	
Evaluator:	InfoGainAttributeEval	(default arguments)
Method:	10 fold cross validation	(list only features with average merit > 0)

average merit	average rank	attribute
0.07 +- 0.001	1.1 +- 0.3	2 word_amnt
0.069 +- 0.001	1.9 +- 0.3	7 da_vfrms
0.058 +- 0.001	3 +- 0	3 da_dur
0.028 +- 0.001	4 +- 0	1 dag_label
0.018 +- 0.001	5 +- 0	8 da_vf_r
0.015 +- 0	6 +- 0	9 da_vf_ps
0.014 +- 0.001	7 +- 0	11 da_i_sd
0.011 +- 0.001	8 +- 0	4 da_w_l_avg
0.004 +- 0	9.4 +- 0.49	5 da_p_mean
0.004 +- 0.001	9.6 +- 0.49	6 da_p_sd
0.002 +- 0.001	11.3 +- 0.46	13 it600_p_sd_d
0.001 +- 0	12.7 +- 0.64	14 it600_vfr_d
0.001 +- 0	13.7 +- 2.79	15 it600_i_mean_d
0 +- 0	15.3 +- 1.27	23 it200_p_sd_d
0 +- 0	16.5 +- 0.67	22 it200_p_mean_d
0 +- 0	18.1 +- 1.14	26 it200_i_sd_d
0.001 +- 0.001	18.4 +- 5.31	10 da_i_mean
0 +- 0	18.7 +- 0.64	24 it200_vfr_d
0 +- 0	18.8 +- 4.26	21 it400_i_sd_d
0 +- 0.001	19.2 +- 3.87	12 it600_p_mean_d
0 +- 0	19.3 +- 0.9	25 it200_i_mean_d
0 +- 0.001	19.3 +- 4.43	17 it400_p_mean_d
0 +- 0	21.6 +- 0.8	16 it600_i_sd_d
0 +- 0	23.9 +- 2.21	18 it400_p_sd_d
0 +- 0	24.4 +- 1.2	19 it400_vfr_d
0 +- 0	24.8 +- 0.98	20 it400_i_mean_d

Appendix D4 Experiment 3.4

Feature sets: Turn + 200ms + 400ms

Goups: backchannel (Ba) non-backchannel (NBa)

Classifier: J48 decision tree (default arguments)

Traning / testing: 10 fold cross validation

Dataset	Correct classifications	# Instances	Precision	Recall	f-Measure
All combined Instances:	64.0%	Ba: 8461 NBa: 8526	Ba: 0.628 NBa: 0.653	BA: 0.677 NBa: 0.602	Ba: 0.652 NBa: 0.627

Feature merit ranking

Dataset: All: ES2002 ~ TS3012

Evaluator: InfoGainAttributeEval (default arguments)

Method: 10 fold cross validation (list only features with average merit > 0)

average merit	average rank	attribute	average merit	average rank	attribute
0.07 +- 0.001	1.1 +- 0.3	2 word_amnt	0 +- 0	12.8 +- 1.4	20 it200_i_mean_d
0.069 +- 0.001	1.9 +- 0.3	7 da_vfrms	0 +- 0	13.1 +- 0.54	21 it200_i_sd_d
0.058 +- 0.001	3 +- 0	3 da_dur	0 +- 0	13.5 +- 0.81	19 it200_vfr_d
0.028 +- 0.001	4 +- 0	1 dag_label	0 +- 0.001	13.8 +- 1.83	12 t400_p_mean_d
0.018 +- 0.001	5 +- 0	8 da_vf_r	0 +- 0	16 +- 3.69	18 it200_p_sd_d
0.015 +- 0	6 +- 0	9 da_vf_ps	0 +- 0.001	16.1 +- 4.72	10 da_i_mean
0.014 +- 0.001	7 +- 0	11 da_i_sd	0 +- 0	16.2 +- 0.4	14 it400_vfr_d
0.011 +- 0.001	8 +- 0	4 da_w_l_avg	0 +- 0	17.2 +- 0.4	13 it400_p_sd_d
0.004 +- 0	9.4 +- 0.49	5 da_p_mean	0 +- 0	18.7 +- 2	15 it400_i_mean_d
0.004 +- 0.001	9.6 +- 0.49	6 da_p_sd	0 +- 0	19 +- 1.1	17 it200_p_mean_d
			0 +- 0	19.6 +- 1.36	16 it400_i_sd_d

Appendix D5 Experiment 3.5

Feature sets: Turn + 200ms

Goups: backchannel (Ba) non-backchannel (NBa)

Classifier: J48 decision tree (default arguments)

Traning / testing: 10 fold cross validation

Dataset	Correct classifications	# Instances	Precision	Recall	f-Measure
All combined Instances:	64.5%	Ba: 8461 NBa: 8526	Ba: 0.646 NBa: 0.643	BA: 0.634 NBa: 0.655	Ba: 0.64 NBa: 0.649

Feature merit ranking

Dataset: All: ES2002 ~ TS3012

Evaluator: InfoGainAttributeEval (default arguments)

Method: 10 fold cross validation (list only features with average merit > 0)

average merit	average rank	attribute
0.07 +- 0.001	1.1 +- 0.3	2 word_amnt
0.069 +- 0.001	1.9 +- 0.3	7 da_vfrms
0.058 +- 0.001	3 +- 0	3 da_dur
0.028 +- 0.001	4 +- 0	1 dag_label
0.018 +- 0.001	5 +- 0	8 da_vf_r
0.015 +- 0	6 +- 0	9 da_vf_ps
0.014 +- 0.001	7 +- 0	11 da_i_sd
0.011 +- 0.001	8 +- 0	4 da_w_l_avg
0.004 +- 0	9.4 +- 0.49	5 da_p_mean
0.004 +- 0.001	9.6 +- 0.49	6 da_p_sd
0 +- 0	12.4 +- 0.49	16 it200_i_sd_d
0 +- 0	12.6 +- 0.49	15 it200_i_mean_d
0 +- 0	13 +- 2	14 it200_vfr_d
0.001 +- 0.001	13 +- 2	10 da_i_mean
0 +- 0	14.4 +- 0.8	13 it200_p_sd_d
0 +- 0	15.6 +- 0.8	12 it200_p_mean_d

Appendix D6 Experiment 3.6

Feature sets: Turn

Goups: backchannel (Ba) non-backchannel (NBa)

Classifier: J48 decision tree (default arguments)

Traning / testing: 10 fold cross validation

Dataset	Correct classifications	# Instances	Precision	Recall	f-Measure
All combined Instances: 16987	64.5%	Ba: 8461 NBa: 8526	Ba: 0.648 NBa: 0.642	BA: 0.629 NBa: 0.66	Ba: 0.638 NBa: 0.651

Feature merit ranking

Dataset: All: ES2002 ~ TS3012

Evaluator: InfoGainAttributeEval (default arguments)

Method: 10 fold cross validation (list only features with average merit > 0)

```
average merit   average rank    attribute
0.07  +- 0.001     1.1 +- 0.3     2 word_amnt
 0.069 +- 0.001     1.9 +- 0.3     7 da_vfrms
 0.058 +- 0.001     3   +- 0       3 da_dur
 0.028 +- 0.001     4   +- 0       1 dag_label
 0.018 +- 0.001     5   +- 0       8 da_vf_r
 0.015 +- 0        6   +- 0       9 da_vf_ps
 0.014 +- 0.001     7   +- 0      11 da_i_sd
 0.011 +- 0.001     8   +- 0       4 da_w_l_avg
 0.004 +- 0        9.4 +- 0.49    5 da_p_mean
 0.004 +- 0.001     9.6 +- 0.49    6 da_p_sd
 0.001 +- 0.001    11   +- 0      10 da_i_mean
```