Cue Phrase Selection Methods for Textual Classification Problems

J.H. Stehouwer
Master of Science Thesis
Human Media Interaction

Research Group Human Media Interaction
Faculty of Computer Science
University of Twente
Enschede, The Netherlands

Graduation Committee:
Dr. Ir. H.J.A. op den Akker
Dr. D.K.J. Heylen
Ir. R.J. Rienks

August 2006

Samenvatting

De classificatie van teksten en delen van teksten, maakt voor een belangrijk deel gebruik van het voorkomen van woorden en combinaties van woorden. Niet elk woord of woordcombinatie speelt een even duidelijke rol als indicator van de classificatie van een stuk tekst. Er is onderzoek gedaan naar methodes voor het selecteren van de meer indicatieve (opeenvolgende) woorden uit de verzameling van woorden en woordgroepen zoals deze in de teksten voorkomen. Deze, meer indicatieve, (combinaties van) woorden noemen we cue-phrases. Het doel van deze methodes is dan ook het eerst selecteren van de meest indicatieve cue-phrases. De geselecteerde verzameling woorden en groepen van woorden kan vervolgens worden gebruikt bij het trainen van het classificatiesysteem. Om deze methodes te bestuderen zijn een aantal experimenten uitgevoerd op een corpus van recepten uit een kookboek, en op een corpus van vergaderingen met vier deelnemers. Teneinde dit te bewerkstelligen is er ook een computer programma geschreven, hetgeen gebruikt is voor deze experimenten.

Voor de experimenten op het corpus van recepten is er gekeken of het mogelijk was de zinnen van recepten te classificeren in verschillende types. Denk hierbij aan een classificatie in types zoals onder andere “requirement” en “instruction”. Voor de experimenten op het corpus van vergaderingen met vier deelnemers is gekeken of het, met enkel deze features, leerbaar is of een zin aan een individu of een groep gericht is. De persoon aan wie een uitspraak gericht wordt wordt ook wel de geadresseerde (Eng. addressee) genoemd.

De experimenten op de recepten leverden goede resultaten op, waarbij duidelijk wordt dat, een aantal van de in dit verslag besproken methodes geschikt zijn om woorden of woordcombinaties mee te selecteren. De experimenten op het vergaderingen corpus waren minder succesvol qua het volbrengen van de classificatietaak. Wel zagen we vergelijkbare patronen qua de prestaties van de verschillende selectie methodes. Gezien de resultaten van Jovanovic kunnen we concluderen dat er voor deze classificatie meer nodig is dan enkel woorden.

Een oorspronkelijk doel was om naar “addressing” in discussies te kijken, komen individueel geadresseerde zinnen vaker voor binnen een discussie? Hiervoor is het echter nodig om eerst de discussiesegmenten te identificeren. Het bleek moeilijk deze betrouwbaar te specificeren en onze initiële definitie schiet dan ook te kort.

Summary

The classification of texts and pieces of texts uses the occurrence of, combinations of, words as an important indicator. Not every word or each combination of words gives a clear indication of the classification of a piece of text. Research has been done on methods that select some words or combinations of words that are more indicative of the type of a piece of text. These words or combinations of words are selected from the words and word-groups as they occur in the texts. These more indicative words or combinations of words we call “cue-phrases”. The goal of these methods is to select the most indicative cue-phrases first. The collection of selected words and/or combinations thereof can then be used for training the classification system. To test these selection methods, a number of experiments has been done on a corpus containing cookbook recipes and on a corpus of four-participant meetings. To perform these experiments, a computer program was written.

On the recipe corpus we looked at classifying the sentences into different types. Some examples of these types include “requirement” and “instruction”. On the four-person meeting corpus we tried to learn, using only lexical features, whether a sentence is addressed to an individual or a group.

The experiments on the recipe corpus produced good results that showed that, a number of, the used cue-phrase selection methods are suitable for feature selection. The experiments on the four-person meeting corpus where less successful in terms of performance off the classification task. We did see comparable patterns in selection methods, and considering the results of Jovanovic we can conclude that different features are needed for this particular classification task.

One of the original goals was to look at “addressee” in discussions. Are sentences more often addressed to individuals inside discussions compared to outside discussions? However, in order to be able to accomplish this, we must first identify the segments of the text that are discussions. It proved hard to come to a reliable specification of discussions, and our initial definition wasn’t sufficient.

Preface

Science is what we understand well enough to explain to a computer.
Art is everything else we do.

Donald Knuth

Five years ago I started studying Computer Science at the University of Twente. In these five years I learned a lot of things, not all of them directly related to Computer Science. As time went by the subject itself became more and more interesting, and I chose to specialise in Human Media Interaction, specifically the Machine Learning part thereof.

When I started on my final projects all kinds of ideas where up in the air, and I started by looking at the final thesis of van der Weijden and the thesis of Kats. Because of the work of van der Weijden I started looking at discussions in meetings, specifically at what discussions are.

The work of Kats and the work of Verbree turned me to the automatic selection of cue-phrases, a method suitable for text-classification problems where the problem can be solved using lexical features. To experiment with these cue-phrase selection methods a tool was made to run the experiments with.

One of the text classification problem I looked at is the recipe-classification task. Another classification task I learned of via Natasa Jovanovic and Rieks op den Akker. This task is concerned with the learn-ability of the addressee of sentences in meetings. Specifically to see if it was learnable whether an utterance is single or group addressed. A lot of time was spend on this problem.

I would like to thank several people for their contribution to this thesis. First of all I want to thank my parents and my little sister, who kept me motivated when things didn't go as fast as I'd like. I want to thank my supervisors, especially Rieks, for the generous amount of usefull and timely feedback given during the whole process. And finally I wish to thank the Student Network Twente and Stichting Abunai for keeping me usefully distracted when I needed to get my mind of things.

All in all it has been a fun five years. In fact five years may be a little too fast to get out of here as there is so much left to learn.

Herman Stehouwer
Enschede, July 2006

Contents

CHAPTER 1

Introduction

It is not unusual these days to spend a lot of time in meetings. While these meetings themselves cost quite a bit of time, the time spend preparing for the meetings and time spend processing the results of the meeting is far greater. A significant amount of scientific work has been done on automating parts of these processes. The work presented in this thesis aims to aid this process, or at least provide some insights.

In the University of Twente itself, a good amount of recent research has focussed on parts of the Augmented Multi-party Interaction project, or AMI for short. The AMI project aims to 'augment' meetings by giving all participants to a meeting valuable and important information about that meeting in a convenient way. This information should help the comprehension of the meeting, streamlining it. This should be done automatically and without user intervention. The corpus of the AMI project consists of a set of role-played meeting augmented with some annotations, including (amongst others) dialogue acts, topic segmentation, gestures, gaze, addressee and dialogue act relations.

One of the topics that has been looked at recently is the visualisation of the structure of meetings, or more specifically the visualisation of discussions in meetings. For this van der Weijden designed a discussion annotation and visualisation method called Twente Argument Schema or TAS [vdW05, RR06b]. For a short description of TAS see appendix E. Some research has been done about the usability of TAS and the results presented in [RR06a] suggest that the application of TAS does help with the comprehension of the meeting structures.

However, the definition of what a discussion actually is, is unclear. In chapter 2 we will look at this issue. We specifically look if there is agreement on the annotation from different annotators given our definition. Of these results a qualitative analysis is done using, amongst other things Adjacency Pairs and Dialogue Acts.

Some words or combinations of words are indicative of the class of a piece of text given a classification problem, one word combination of words more so than the other. These combinations cue us to a certain class, hence the name cue-phrases. As Samuel, Carberry and Vijay-Shanker show in [SCVS99] cue-phrases are one of the most effective features when used for dialogue act tagging. It is likely that the methods presented in this paper generalise to other tasks than dialogue act tagging and to other corpora. Therefore we will discuss their cue-phrase selection methods in chapter 3 in detail.

Quite a bit of work has been done recently at the university that involves the automatic selection of cue-phrases to use as features while learning. Both Kats and Verbree used automatically selected cue-phrases in their research. The goal of cue-phrase selection methods is to select the most indicative (combinations of) words first. So a cue-phrase selection method is, for a certain classification task, better than another if the achieved accuracy rises faster in the beginning. It is also desirable to have computationally efficient cue-phrase selection methods.

In [Kat06] Kats used the cue-phrase selection methods from [SCVS99] on the IMIX sentence classification task. Verbree used a few of his own selection methods in [Ver06] on the TAS node sentence classification task. The methods used by both Kats and Verbree will be discussed in chapter 3. Using cue-phrase selection methods one can select appropriate “cues” and use those for learning. These cues are lexical cues, and therefore suitable for classification tasks where lexical features are adequate by themselves, or help the performance of the classification. In order to facilitate the experimentation with these different cue-phrase selection methods on different classification tasks and corpora a tool was written, in the chapter on cue-phrases this tool is also presented.

The first classification task on which the cue-phrase selection methods are tested is the recipe corpus. The recipe corpus, containing sentences from cook-book recipes, and the corresponding classification task are presented in chapter 4. The recipe classification task entails that each sentence is assigned a certain type, such as requirement, name, instruction blocking, etc.

The second task on which the cue-phrase selection methods are tested is that of the identification of the addressee of an utterance. Specifically to classify whether an utterance is addressed to an individual or to a group. Addressing is an important part of communication [Gof81, NJN06] and helps disambiguate discussions in meetings. Jovanovic has produced excellent results in [NJN06] on identifying the addressee of an utterance. She finds out whom exactly the utterance is addressed to, a task more complex than our single or group addressed task. However for this she uses a number of computationally expensive features such as gaze and gesture information. The details of this task are explained in chapter 5. We only use the selected cue-phrases in most of the experiments, but also run a smaller series of experiments using dialogue acts and also the length of the utterance.

The main goal is to look at the different cue-phrase selection methods on two different classification tasks. Also we are interested to see if we can classify if: A sentence is addressed to a single person or a group, based on the selected cue-phrases, as this is a computationally cheap feature.

CHAPTER 2

Discussions

Previous work done by van der Weijden (in [vdW05]) produced a scheme for annotating discusions called TAS that aims to get a more detailed view of the "structure" of those discussions. TASs is discussed in detail in appendix E. TAS is applied to discussion segments, but how are those segments determined? Do human annotators agree on this? This chapter aims to look briefly at this issue and provide some footholds for future research. Discussions are an important part in meetings, as they are a means to settle differences in opinion.

An example of a discussion, as found in meeting IS1003c is given in figure 2.1. As can be seen participant C thinks it is an advantage of the intelligent controller that it can have voice recognition. Opposed to Participant C are in this, rather short, discussion participants A and D who think voice control would be a disadvantage.

This example conforms to the definition of discussion as found in [Rie06]: *"A dialogue,* **related to a single statement or proposition***, on which not all participants agree."* (emphasis mine). What we want to know is: *Is this definition complete enough for discussion annotation and what kind of features could be used to learn these annotations?*. To see whether this definition alone is enough to provide a reliable annotation an annotation manual was created and discussions where annotated. The exact method used is explained in the next section.

```
C: Yeah , thats thats the advantage of intelligent controller . Even you h you have the controller S
, I can [other] I can say channel three , so its c come to channel three , I dont have to [laugh]
.[disfmarker]
D: No .
B: [laugh] Its [disfmarker] its [gap] [laugh]
D: No , but this is disadvant disadvantage .
A: Yeah , I think its a disadvantage .
C: Its advantage .
```

Figure 2.1: *An example of a discussion as found in meeting IS1003c.*

2.1 Method

In order to answer the question we must first see whether different annotators can use this definition of discussions. In other words whether the annotators agree on the location of discussions as they are defined by this definition. It was decided that we would annotate discussions IS1003a, IS1003b, IS1003c and IS1003d from the AMI corpus. To do that an annotation manual was written. A tool called ArgumentA was used to annotate the discussions. ArgumentA was also used to mark discussions for TAS annotation and it was written by van der Weijden for his final project (see [vdW05]). The annotation manual that was written can be found in appendix B. With the results of this annotation we can then try to answer the questions.

The data collected consists of three annotators marking the location of all discussions in meetings IS1003a, IS1003b, IS1003c and IS1003d. The start and end points of all discussions where marked on the transcription, colouring the lines where the discussion-annotation denoted an end or a start of a discussion. These marked transcriptions are not included in this thesis because of large amount of pages they take up.

Due to the small amount of annotation data available, a quantitative analysis would not produce reliable results. So, a qualitative analysis, aided by a small collection of scripts, was performed.

The data that was available was used to extract some simple statistics. One of these produces for the start- and end-points of discusions several datapoints. These datapoint include the label the annotator marked the discussions with, the dialogue of the starting/ending utterance, the addressee of the dialogue act, the speaker of that dialogue act and the complete utterance that belongs to that dialogue act. This data is available in a tabular format in appendix C. In section 2.2 (below) we discuss the Dialogue Acts with which the AMI corpus is annotated.

To see whether there is a difference in the distribution of dialogue acts and addressing inside discussions compared to outside discussions information was compiled about this. Should these differences be large they could be exploited for use in automatic annotation of these discussions.

The number of dialogue acts inside and outside discussions where counted for each annotator for each discussion. These can be found in appendix 2.3. The distribution of Dialogue Acts inside and outside of discussions was also counted, these can be found in table 2.5. Finally words occurring inside and outside discussions in a meeting where counted for all three annotators for all meetings. The results of this will be discussed below, but the raw data will not be given as it would take up a lot of space. This was stored in one file per annotator, one file per discussion.

We counted the number of group addressed Dialogue Acts compared tot the total number of dialogue acts inside and outside discussions for meetings IS1003b and IS1003d for each annotator, these numbers can be found in table 2.4. We looked at the distribution of the Adjacency Pairs and addressing inside and outside discussions for meetings IS1003b and IS1003d for all annotators, these distributions can be found in appendix D. These two pieces of information are only available for discussions IS1003b and IS1003d because those two discussions are the only meetings annotated with our annotation manual for discussion annotation that also have the addressee annotation available.

The results obtained in this way will be discussed below in section 2.3.

2.2 Dialogue Acts

As we look at the AMI dialogue acts at the start and end of discussions, as well as their distribution inside and outside discussions, they will be discussed here briefly. For a detailed description please see [AMI]. Dialogue Acts are a way of marking a transcription according to speaker intention.

An essential part of dialogue act coding is the determining where the boundaries between the different DA's lie. It is quite possible to have a DA span multiple utterances, for example "No, it's not. It's not.". As well as that a single utterance may be split in multiple segments if necessary. For all the different DA-labels discussed below there are several examples as well as clear instructions to prevent common confusions in [AMI]. Now on to the DA's:

Backchannel Short utterance in the background that doesn't really stop the current speaker. Typically used to communicate to the current speaker that the speaker of the backchannel is still listening.

Stall A short utterance, used when the speaker hasn't figured out what to say, or to try to get or keep the attention of the group. Note that **Stall** itself isn't really a dialogue act, just a convention used in the AMI project, since it is usefull to label every utterance, even those not really conveying intention (besides the 'I want to keep speaking' part).

Fragment A fragment is a segment that isn't a **Stall**, nor a **Backchannel** but doesn't convey speaker intention.

Inform Gives information.

Elicit-Inform Requests information from someone else.

Suggest Suggests a course of action.

Offer The speaker expresses an intention of his/her own actions.

Elicit-Offer-Or-Suggestion The speaker wants somewhone to make a suggestion or an offer.

Assess Expresses an evaluation ('That would be good', 'yeah', 'it is a big number', etc.).

Comment-About-Understanding A DA for segments that indicate wether the previous speaker was understood.

Elicit-Assessment A DA for segments that (tries to) elicit an assessment from a speaker.

Elicit-Comment-About-Understanding The speakers elicits a responce from a (group of) speakers about wether the speaker was understood.

Be-Positive A social act that is intended to make an individual or the group happier.

Be-Negative A social act that makes an individual or the group less happy.

Other Any segment that doesn't fit with any of the above DA's. For example mumbling aloud, 'Now, where was I?'.

AMI annotation also contains a relation annotation linking an utterance to another utterance. This is only done if the annotator thinks a relation exists. It only contains a few relations, namely:

Positive The target supports the intention of the source. For example (taken from [AMI]):

```
Positive Relation Example
        ...upon picking up a whiteboard pen and stepping up to the white-
        board for the first time...
C       Okay | VERY NICE | alright
```

Negative The target rejects the source. For example (taken from [AMI]):

```
Negative Relation Example
A       Mm. So, some kind of idea uh with um um cel lular phone with
        a a screen that wil l tel l you what, no .
C       NO, NO SCREENS | its too complex.
```

Partial The target partially supports but partially rejects the source. For example (taken from [AMI]):

```
Partial Relation Example
[C is drawing on the whiteboard]
D       A kind of snake? A cobra?
C       Yeah, uh | NOT REALLY, | a small cobra.
```

Uncertain The target expresses uncertainty about the source. For example (taken from [AMI]):

```
Uncertain Relation Example
C       We can adapt only one switch, suppose here like we can make
        two switches and if Im left-hander I use this switch to follow
        the main operations.
B       I mean if its less than three uh then we can make it uh like a...
D       THREE BUTTONS, YOU MEAN?
```

2.3 Results

To evaluate, by hand, the degree of annotator agreement we look at the transcriptions marked with the start- and end-points of discussions. As stated before the discussions were annotated for all four meetings by three different annotators which we shall refer to as H (for Herman), J (for Jan) and R (for Rieks). Looking at the transcriptions it quickly becomes clear that annotators H and R seem to agree about the position on a decent amount of the discussions, though almost always with start and end utterances that lie a few utterances apart. In meeting IS1003d the disagreement between all annotators is quite large. It is quite likely that the reason why H and R agree some discussions is that, prior to the writing of the annotation manual, they discussed the nature of discussions in some detail, thus creating some sort of agreement. One example of such a discussion (from the IS1003b transcription) can be seen in table 2.1.

Looking at this example we see R and H agree that this is a discussion, however their start-points for this discussion lie some utterances apart. This is often seen in the transcription and R and H rarely agree on the exact start or end of a discussion (though they do agree on the end in this particular case).

Sentence	H	J	R
C: And just to have uh an idea , do you think you as the			S
C: User Interface Designer to would it be possible to have less buttons and still have the same functionality and to have powerful remote control , you think it's possible ? Sure ? Yeah ?			
D: Yeah .			
D: Yeah , I think possible . Because we can [disfmarker]			
D: We can uh			
D: mix uh several function in one button .			
C: Yeah .			
D: So lets you [disfmarker] then you have less buttons . But I'm not sure [disfmarker]			
C: Yeah , but do you think it will be easy to use ? Because if you have many functions just for one button it would be quite difficult for the user to know .	S		
A: Yeah , remember the user is not happy to read the			
C: Yeah , I think the [disfmarker]			
B: The manuals .		S	
A: manual . It's [disfmarker]			
D: No you you can have a switch menu , so you can			
C: Yeah , but it has to be intuitive .			
D: well for example [disfmarker]			
D: Yeah , I think so . Like for for example you can uh you can category the function i i into several classes . Then			
D: for um you can have a switch menu , so you put the switch menu to it it tend to this kind of this category of functions . Then you you put the switch button , then it			
C: Yeah , okay .			
C: Okay , but [disfmarker]			
D: switch to another category of functions .			
D: Yeah .			
D: For example , if you have remote control you you can rem you can control your T_V_ and also you can control your uh recorder .			
B: With a [disfmarker]			
D: So there's a different functions , but i if you you [disfmarker] there's a button you can switch			
between control T_V_ and control your recorder .	E		E

Table 2.1: *An example of a discussion in which the Hand R agree on the discussions, but have different starting points. Taken from the IS1003b meeting.*

Sentence	H	J	R
D: No y you do the minutes first , or ?		S	
A: What ?			
D: No ?			
A: I I think I will let uh our User Interface Designer speak first , Mister David Jordan .			
D: Okay .			
C: Yep .		E	

Table 2.2: *An example of something of which annotator J believes that it is a discussion, but H and R do not. Taken from IS1003c.*

Very often it is seen that only annotator J thinks that there is a discussions, but that annotators H and R don't agree with this. One such example (from discussion IS1003c) can be found in table 2.2.

J mostly sees discussions, when the others do not, when there is not a difference of opinion, but a difference in beliefs of the participants. Such a difference of beliefs is often quickly resolved and mainly involves synchronizing information between the participants of the meeting. We, on the other hand, would like to have the differences of opinion annotated.

So we can state about these results that the overall inter-annotator agreement is poor. That, even though they agree on the nature of discussions, H and R still disagree most of the time about the exact start and end of discussions. H and R seem to agree about the position of most discussions, except in meeting IS1003d. This is also supported by the numbers shown in table 2.3 about the number of utterances inside and outside of discussions.

Appendix C contains tables listing, for the annotated beginnings and endings of discussions, the corresponding AMI Dialogue Act and information related to that Dialogue Act (the addressee, the speaker, the complete utterance belonging to that DA).

From this, automatically extracted, data it becomes clear that the segmentation as used by TAS and the segmentation as used by the AMI Dialogue acts is quite different. For example the second discussion as annotated by J in IS1003a starts with a “Stall” Dialogue Act belonging to the utterance *“So”*, the corresponding TAS segment is *“So the selling price of the product will be twenty five Euros . [laugh] Yeah . Yeah . [other] I S think its quite good price , yeah . And uh”*. The start and end of discussions were annotated using the Argumenta tool, which used the TAS segmentation.

From the same data it also becomes clear that discussions tend to start on a Stall (35%), Suggest (11%), Asses (10%) , Inform (15%) , Elicit-asses (12%) or Elicit-inform (9%) while most discussions end with a Stall (13%), Backchannel (40%)), Comment-About-Understanding (5%), Asses (27%) and sometimes an Inform (10%). Related to this are the tables of the distribution of Dialogue Acts inside and outside discussions in the annotated meetings, these can be found in table 2.5. For example from the table for meeting IS1003c and annotator H, for which around 50% of the meeting is marked as discussion, we see for example a higher number of the “suggest” dialogue act ouside discussions. There are some differences between the dialogue act distributions inside and outside the annotated discussions for all annotators, with some DA’s being more common inside and others more common outside of discussions. While clear, these differences are not that large for the frequently occuring DA’s.

When looking at the numbers of words inside and outside of discussions no words which occur much more inside the discussions where found. Of the list of words, words that only occurred within discussions, or which occurred outside discussions infrequently, no words occurred frequently in the meeting. For example, though the word *switch* only seems to occur within discussions in meeting IS1003b it occurs only 6 times in the complete meeting. Most, around 95%, words that only occur within discussions, or only once or twice outside discussions occur in dicussions only 3 or fewer times. So no reliable indicators of discussions within the AMI meetings IS1003a/b/c/d where found.

Some data was also compiled which is only available for meetings IS1003b and IS1003d. The first set of this is the amount of group addressed utterances inside and outside of discussions, compared to the number of DA’s inside and outside discussions. This data can be found in table 2.4 compared to table 2.3. We can see in the data that the amount of group addressed DA’s is, proportionally, larger outside discussions (around 30%) than inside discussions (around 20%). However less than 30% of the total DA’s is group addressed, so the difference in the proportion of group-addressed DA’s can give no more than a possible indication of a discussion. The same, slight, differences are also in the second set of data, that of the speaker patterns of the annotated Adjacency Pairs, which can be found in Appendix D. However these differ a lot between IS1003b and IS1003d and between annotators also. Conclusions cannot be drawn.

2.4 Conclusion and Discussion

Concluding it can be said that the current definition is not clear enough to achieve a consequent annotation. The definition needs to be further clarified at the very least. From the small amount of data that was collected it became clear that the annotators disagreed on where the participants disagreed. All annotated discussions contained some form of disagreement, though also of dif-

		H	**J**	**R**
IS1003a	Inside	45	127	50
	Outside	373	290	368
IS1003b	Inside	169	357	269
	Outside	524	336	424
IS1003c	Inside	446	456	494
	Outside	485	475	437
IS1003d	Inside	885	1080	306
	Outside	678	483	1257

Table 2.3: *The number of dialogue acts contained in discussions and the number of dialogue outside discussions for all annotated meetings.*

		H	**J**	**R**
IS1003b	Inside	29	80	58
	Outside	148	97	119
IS1003d	Inside	247	308	81
	Outside	247	114	341

Table 2.4: *The number of group addressed dialogue acts contained in discussions and the number of group addressed dialogue outside discussions for all annotated meetings. Note that for the discussion-annotated meetings addressee information is only available for meetings IS1003b and IS1003d.*

da	**IS1003a**						**IS1003b**						**IS1003c**						**IS1003d**					
	H		**J**		**R**		**H**		**J**		**R**		**H**		**J**		**R**		**H**		**J**		**R**	
	i	**o**	**i**	**o**	**i**	**o**	**i**	**o**	**i**	**o**	**i**	**o**	**i**	**o**	**i**	**o**	**i**	**o**	**i**	**o**	**i**	**o**	**i**	**o**
Assess	9	40	91	76	9	40	31	62	45	48	42	51	75	92	91	76	88	79	138	102	178	62	57	183
Backchannel	4	19	42	56	5	18	11	59	35	35	24	46	36	62		98	51	47	65	79	83	61	28	116
Be-Positive		3		2		3	2	6	6	2	3	5	2			2		2	25	42	19	48	8	59
C-A-U	2	22	15	16	3	25	14	16	14	16	12	18	17	14	15	16	16	15	39	9	39	9	9	39
Elicit-Assessment		7	8	14		7	5	5	3	7	7	3	9	13	8	14	7	15	13	9	19	3	5	17
E-C-U		1		2		1		8	3	5	1	7		2		2		2						
Elicit-Inform	2	22	16	34	2	22	5	23	16	12	7	21	18	16	16	18	21	13	29	14	39	4	8	35
E-O-O-S		2	6	14		2	1	3	3	1	2	1	4	10	6	8	8	6		2		2		2
Fragment	5	71	86	69	5	71	28	57	48	37	49	36	91	64	86	69	89	66	185	123	215	93	65	243
Inform	17	97	88	111	18	96	31	160	104	87	64	127	95	104	88	111	105	94	207	134	269	72	76	165
Offer		5	1	7		5	4	7	4	7	3	8	2	6	1	7	2	6	1	3	1	3		4
Other	1	16	7	7	2	17	1	11	5	6	1	10	7		7		1	6	32	29	32	29	5	56
Stall	5	46	53	59	6	40	17	56	36	37	26	47	59	53	53	59	67	45	107	105	137	75	33	179
Suggest		23	43	37		23	19	52	35	36	28	43	31	49	43	37	39	41	44	27	49	22	12	59

Table 2.5: *The distribution of dialogue acts inside and outside discussions for all four annotated meetings and all three annotators. "i" marks columns denoting the distribution inside discussions, "o" marks columns denoting the distribution outside discussions.*

ferent sorts. The discussions as annotated are continuous segments, it could also happen that a discussion is “split up” by some irrelevant material in between. It is undefined how the annotators should deal with this, but annotators can either annotate two separate discussions, or annotate it as one long discussion. A clear, and rather extreme, example of a difference of this sort occurs between annotators H and R in the first half of meeting IS1003d. For the one discussion annotated by R, H annotated six different discussions.

Annotators disagree on the exact start and end of discussions most of the time, though frequently, if they agree on the discussion in general, these differ only a few utterances. The difference between annotator J and annotators H and R seems to be primarily about the type of disagreement needed. To improve the definition it can be pointed out that the disagreement should present a difference of opinion, not a difference of beliefs. While a difference of beliefs is defined as a discussion by the used definition, it isn’t the type of disagreement that we would like to have annotated. A new, slightly improved definition could, for example, be: *“A dialogue, related to a single statement or proposition, on which not all participants agree. The disagreement should be a difference of opinion, and not a difference in beliefs about the world.”*. With this new definition, and a much improved annotation guide a new experiment should be performed as test. We unfortunately lack the time.

It turned out that there was a certain pattern to the start and end of discussions as far as Dialogue acts is concerned. Some of the Dialogue Acts appeared a lot at the start and/or end of discussions. This could be used as a feature for learning. Other spotted differences could possibly aid the identification of discussions but occur too infrequently to be of help for the automatic identification of discussions in meetings. Galley and others looked at the learnability of agreement/disagreement in [MGS04]. They looked at structural, duration and lexical features. These features could be interesting for further research. Other interesting features are low level features such as F0, speaker overlap and talking speed. Interesting higher-level features could be adjacency pairs, topics (the AMI corpus already contains topic segmentation annotation) and the grammatical type of an utterance. However without annotator agreement on discussions and without a precise definition, not to mention more annotated data, we cannot do anything but confine ourselves to hypothesize.

CHAPTER 3

Cue Phrases and MAW

Some combinations of words are indicative of the class of a piece of text given a classification problem, one combination more so than the other. These combinations cue us to a certain class, hence the name cue-phrases. There has been quite a bit of previous work done on cue-phrases, including the work of Kats and the work of Verbree, which is discussed in section 3.2, and the work of Samuel et al. ([SCVS99]) which was also used by Kats in [Kat06] and which is discussed in section 3.1. This work discusses cue-phrase selection methods. These methods have to goal of selecting the most indicative cue-phrases first.

As one will see there are different methods for cue-phrase selection. With this we mean a way to rank all possible cue-phrases in some order in such a way that the first n selected cue-phrases will be more descriptive of one of the classes of the classification which one is trying to learn. This means that the performance gains observed, when one adds more cue-phrases as features, should be much larger for the first few cue-phrases than for the later cue-phrases. This desired behavior is clearly visible in figure 3.1. One cue-phrase selection method is better than another, for a certain task on a certain corpus, if it approaches the maximum accuracy the fastest. With the maximum accuracy being the largest accuracy obtainable using cue-phrases, to this value most selection methods converge.

We will first discuss the cue-phrase selection methods in some detail. Then we will conclude the need to build a computer program to perform experiments with these selection methods, and finally we will describe this tool.

3.1 Cue-Phrase Selection Methods

In *"Automatically Selecting Useful Phrases For Dialogue Act Tagging"* ([SCVS99]) Samuel, Carberry and Vijay-Shanker propose several cue-phrase selection methods. In this section we will discuss the cue-phrase selection methods proposed in that paper. These methods will be used in several experiments on different classification tasks in this thesis.

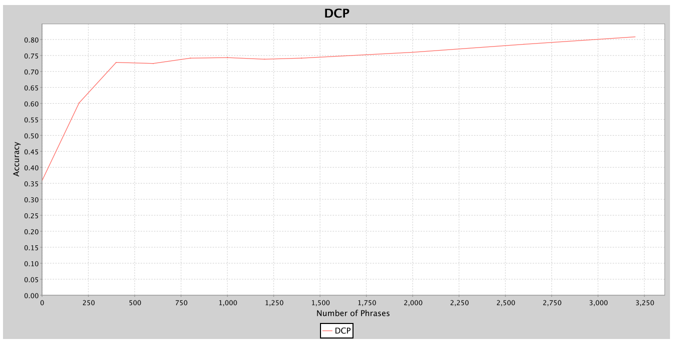


Figure 3.1: *An example of the desired behavior of a cue-phrase selection method. In this case using the DCP method on the Recipe classification task with a J48 learner.*

It is important to note that these methods for cue-phrase selection can be applied to any learning task where cue-phrases can potentially be used for classification. This means that it is applicable to any task where lexical features can be used for learning. In [SCVS99] it is noted that the DCP (Deviation Conditional Probability) cue-prhase selection method outperforms the others. Which is rather unexpected since DCP is a new metric unlike Information Gain or Mutual Information. Samuel et al. used a Transformation Based Learner to achieve their results.

A few definitions which are used in the formulas below will now be listed.

p A possible cue-phrase.

d A target for a cue-phrase, Samuel et al. used cue-phrases for dialogue-act classification.

$\#(x)$ The frequency of an event x.

$P(x)$ The probability of x.

$P(x|y)$ The probability of x given y.

D The number of different targets.

U The number of items in the training set, Samuel et al. used utterances.

$\overline{p}$ The set of utterances in which p does not appear.

We will now discuss all the cue-phrase selection methods as they where used by Samuel et al. ([SCVS99]) and Kats ([Kat06]). In several places we will use the described method on an example possible cue-phrase. For this we will use the following example "corpus":

Sentence	Class
The house is on fire .	A
The cat is not .	B
What a day .	B
The last sentence .	A

COOC [SCVS99] discusses several cue-phrase selection methods, of these the **coocurrence** metric is the simplest. The **cooccurrence** metric simply gives each possible cue-phrase a number equal to the highest cooccurrence it has with one of the targets. In [SCVS99] those targets are dialogue acts. The formula for COOC is as follows:

$$COOC(p) = \max_d \#(p\&d)$$

When using our example corpus and the possible cue-phrase "The" the scores would be as follows. For the target "A" the cooccurrence score would be two, for the target "B" the score would be one, therefore the phrase gets a total score of two.

When using this method possible cue-phrases with a higher score are more likely to be cue-phrases. This means that the cue-phrase that co-occurs with one of the targets the most is the first to be selected. The COOC metric does not take into account the a-priory distribution of the targets it is suppsed to learn. For example if the word "the" occurs 100 times in utterances of type A, and less often in other types, the COOC score of "the" will be 100.

CP Another possible selection method is the **conditional probability** metric. CP is a simple conditional probability of a cue-phrase given a target. The formula for CP is as follows:

$$CP(p) = \max_d P(p|d)$$

When using our example corpus and the possible cue-phrase "the" the scores would be as follows. For the target "A" the CP score would be 1, for the target "B" the CP score would be 0.5, therefore the phrase gets a total score of 1.

When using this method possible cue-phrases with a higher score are more likely to be cue-phrases. Conditional Probability does take into acount the distribution of the targets, however it does not take into account the frequency of the dialogue acts directly. CP selects cue-phrases purely on the change of finding the cue-phrase in a sentence if that sentence is of the type target. For example if the cue-phrase "the" occurs in half the phrases of type A and less in other types, the CP score of "the" will be 0.5.

ENT The **entropy** metric tries to find a cue-phrases that correlate with a single target frequently and infrequently with other targets. The entropy metric is based on the standard, and well-known, definition of entropy as it is covered in many sources such as on page 224 of [JM00]. This is covered by the entropy of the dialogue acts given a phrase:

$$ENT(p) = -\sum_d P(d|p) \log_2 P(d|p)$$

When using our example corpus and the possible cue-phrase "the" the scores would be as follows. For the target "A" the partial score would be

$1 * 0$, for the target "B" the partial score would be $0.5 * -1$, therefore the phrase gets a total score of $-(0 - 0.5) = 0.5$.

When using the ENT metric possible cue-phrases with a lower score are more likely to be cue-phrases. However a cue-phrase that occurs equally likely across all targets will get a good score when using this method. The method will then, incorrectly, flag this cue-phrase as containing usefull information. To eliminate this problem in [SCVS99] they examined different ways of accounting for the dialogue-act distribution. These different metrics use the Kullback-Liebler distance, mutual information, the t-test and information gain.

S The **selectional preference strength** metric is a special case of the **Kullback-Leibler distance**. It considers the difference between the distribution of targets given a possible cue-phrase and the a priory distribution of targets:

$$S(p) = \sum_d P(d|p)[\log_2 P(d|p) - \log_2 P(d)]$$

When using our example corpus and the possible cue-phrase "the" the scores would be as follows. For the target "A" the partial score would be $1 * (0 + 1)$, for the target "B" the partial score would be $0.5 * (-1 + 1)$, therefore the phrase gets a total score of 1.

When using the S metric possible cue-phrases with a higher score are more likely to be cue-phrases. This metric is discussed for instance on MathWorld[1]. It measures the difference between two discrete distributions and gives this distance a number. Note however that it is not a true distance measurement since, as can be seen clearly from the formula and what is also noted on mathword, the distance between p and d is not equal to the distance between d and p.

MI The **mutual information** metric measures the reduction of uncertainty to an utterance's target when the utterance contains the possible cue-phrase:

$$MI(p) = P(p) \sum_d P(d|p)[\log_2 P(d|p) - \log_2 P(d)]$$

When using our example corpus and the possible cue-phrase "the" the scores would be as follows. For the target "A" the partial score would be $1 * (0 + 1)$, for the target "B" the partial score would be $0.5 * (-1 + 1)$, therefore the phrase gets a total score of $\frac{3}{4} * 1 = \frac{3}{4}$.

When using the MI metric possible cue-phrases with a higher score are more likely to be cue-phrases. The Mutual Information metric as given in [SCVS99] closely resembles the selectional preference strength metric. All remarks for that metric are also valid here.

TTEST The **t test** metric measures the statistical difference between two distributions. In this case the difference between the a priori distribution of targets and the distribution of targets given a possible cue-phrase.

[1] `http://mathworld.wolfram.com/Kullback-LeiblerDistance.html`

$$TTEST(p) = \#(\overline{p})\sqrt{\sum_d \frac{D^2 - D}{[D\#(p\&d) - \#(p)]^2 + [D\#(d) - U]^2}}$$

When using the TTEST metric possible cue-phrases with a higher score are more likely to be cue-phrases.

IG The **information gain** metric measures the reduction in entropy of the targets resulting from partitioning utterances based on the possible cue-phrase:

$$IG(p) = \sum_d [P(p)P(d\&p)\log_2 P(d\&p) + P(\overline{p})P(d\&\overline{p})\log_2 P(d\&\overline{p}) - P(d)\log_2 P(d)]$$

When using the IG metric possible cue-phrases with a higher score are more likely to be cue-phrases. Information Gain is a frequently used metric within machine learning, for instance in the weka toolkit ([IHW05]) it is used in various learning algorithms such as the J48 and ID3 tree-based classifiers.

D The **deviation** metric is the first of the two new metric proposed in [SCVS99]. As elaborated in [SCVS99] there are two ways in which a cue-phrase may fail; First of all it may be **unsound**, meaning that the left-to-right rule, **IF cue-phrase THEN target**, applies incorrectly in some cases. Secondly a cue-phrase may be incomplete, meaning that it does not occur in all utterances of the type it is a cue-phrase for. In [SCVS99] a single penalty point is assigned for each unsound and incomplete instance.

The **deviation** metric adds the unsoundness and incompleteness of a possible cue-phrase together. This means that the D method is a measure of the number of times miss-classification would occur when using the cue-phrase as a completely reliable indicator of a class. It counts the number of times the cue-phrase

$$D(p) = \min_{d^*}[\#(\overline{p}\&d^*) + \sum_{d \neq d^*} \#(p\&d)]$$

When using our example corpus and the possible cue-phrase “the” the scores would be as follows. For the target “A” the partial score would be $0 + 1$, for the target “B” the patrial score would be $1 + 2$, therefore the phrase gets a total score of 1.

The score for the best target for this possible cue-phrase is selected. When using the D metric possible cue-phrases with a higher score are more likely to be cue-phrases.

DCP Because, like COOC, the D metric does not take into account the a priori distribution of targets another metric was introduced: **deviation conditional probability**. The **deviation conditional probability** method works in the same way as the **deviation** method, only now while using conditional probability. This means that the DCP method is a measure

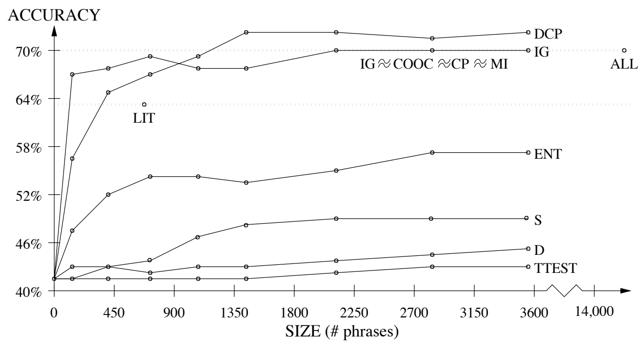


Figure 3.2: *The results for all cue-phrase selection methods from [SCVS99], image from [SCVS99].*

of the chance of miss-classification when using the cue-phrase as a completely reliable indicator of a class. It sums the chance the cue-phrase doesn't occur in a sentence of the correct class and summed with the total chance the cue-phrase occurs in other classes.

$$DCP(p) = \min_{d^*}[P(\overline{p}|d^*) + \sum_{d \neq d^*} P(p|d)]$$

When using our example corpus and the possible cue-phrase "the" the scores would be as follows. For the target "A" the partial score would be $0 + 0.5$, for the target "B" the patrial score would be $0.5 + 1$, therefore the phrase gets a total score of 0.5.

When using the DCP metric possible cue-phrases with a lower score are more likely to be cue-phrases ([SCVS99]). In [SCVS99] this is best performing cue-phrase selection metric.

When testing these cue-phrase selecting methods Samuel, Carberry and Vijay-Shanker found that the DCP method worked best, followed by IG, COOC, CP and MI. The other methods performed poorly. Their results are summed up in figure 3.2. They also used a lexical filter, which is described as:

```
IF a phrase p has a subsequence p that is ranked higher
AND both p and p were selected for the same target
THEN remove p
```

When using this form of lexical filtering they found that filtered-DCP performed no better then regular DCP. However, filtering it converged on the best result faster (while increasing the number of used cue-phrases). They found that the other methods performed slightly better filtered that non-filered. The results for DCP and IG are shown in figure 3.3.

It should be noted that in principle computationally cheap cue-phrase selection methods should be preferred over more expensive ones. The COOC, CP,D and DCP methods are computationally cheap compared to the other methods.

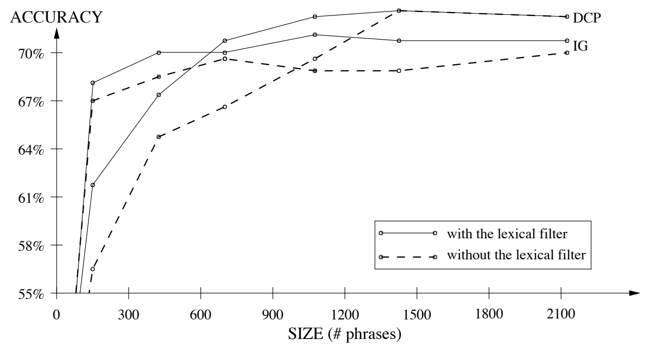


Figure 3.3: *The results for the cue-phrase selection methods DCP and IG from [SCVS99] when used with lexical filtering, image from [SCVS99].*

3.2 The work of Kats and the work of Verbree

Kats

Amongst other features Kats uses automatically selected cue-phrases to determine sentence types for a Question Answer system, constructed as part of the IMIX research program. For this Kats used the cue-phrase selection methods as proposed by Samuel et al. ([SCVS99]), with a few differences. First of all he used regular expressions as cue-phrases instead of n-grams, secondly he used a list of n cue-phrases, one list for each classification target, as a single feature instead of having each cue-phrase as a separate feature. He concludes that the COOC, DCP and IG metrics performed the best if used in this way with lexical filtering turned on.

As we use the cue-phrase selection methods proposed by Samuel et al. ourselves they are discussed below in section 3.1.

Verbree

In his thesis [Ver06] Verbree uses several cue-phrase selection methods to aid in the classification of sentences into TAS node types, TAS is discussed in appendix E. Though he selected n uni-grams, n bi-grams, n tri-grams and n quadri-grams that is irrelevant for the ranking method employed.

Select1/3Normalized is what Verbree calls the first cue-phrase selection method he uses. The S1/3N method selects ngrams only if it matches two conditions. First it must occur at least 3 times in the training set, secondly after normalisation it must have an occurrence ratio which is equal or bigger than $\frac{1}{3}$. With an occurance ratio Verbree means that in more than $\frac{1}{3}$ of the cases it appears in sentences of a certain type *if all sentence types would occur as frequently as the others.*

The second cue-phrase selection method used by Verbree he calls **DROP**n. In this method a cue-phrase is selected if it matches two conditions. First it must occur at least 3 times in the training set. Secondly there must be n sentence types in which the cue-phrase does not occur. In [Ver06] Verbree used DROPn with as values for n, 1 and 2.

The third and final cue-phrase selection method used by Verbree he calls **TOPx**. TOPx ranks the cue-phrases and then selects the top x of them. By selecting only the most predictive cue-phrases a lot of "noisy" cue-phrases are eliminated. The ranking score is based on two assumptions. Firstly Verbree assumes that *"Of two ngrams, the ngram which is more biased (after normalization) to a specific type (...) has a bigger probability of being the cue-phrase than the other."*. Secondly he assumes that cue-phrases which occur more often, if they are equaly biased, are better. His ranking formula is the product of the number of times the cue-phrase occurs in sentences of type X and the part of the 'ngram-space' occupied by nodes of type X.

Conceptually, based on the two assumptions, this resembles CP, with the small difference that CP doesn't take the number of times a cue-phrase occurs into account. However, in practise it is a multiplication of the CP and COOC methods together. Unfortunately the number of times a cue-phrase occurs has a much bigger influence on the final number than the occupation of the "ngram-space". After all the the occurrence number can be infinitely large, while the part of the space cue-phrase occurs in is expressed as a fraction ($[0, 1]$). All in all TOPx remains very biased to often occurring n-grams, selecting those as cue-phrases, for example the word "the" occurs very often in English texts.

Verbree notes that he achieves slightly better results when selecting cue-phrases based on their order. Meaning selecting n cue-phrase uni-grams, n bi-grams, n tri-grams and n quadri-grams. However the order-specific cue-phrase selection method he employs selects more n-grams if the selection is done order-specific. Then again, as shown in [SCVS99] higher-order n-grams can add something even if lower-order n-grams with the same words are already selected, unless those are selected for the same class.

Another technique Verbree explores is that of compression. This means that all cue-phrases indicating a certain class are grouped together as a single feature. For each utterance this feature has as value the number of times a cue-phrase from its list occurs in that utterance. We do not use compression in our experiments as it performs worse than individual cue-phrases.

3.3 Building MAW

When we look at the list of different cue-phrase selection methods we see that, in order to perform a meaningful number of experiments, and to allow others to also perform comparable experiments, we needed a tool. After some investigation it was decided that it was easier to write such a tool from scratch than to adapt the code used by Verbree for his experiments for [Ver06]. The tool reads in a corpus, uses that data to create a training and a testing-set and extracts features from each sentence/utterance/block (corpus dependent). For the learning part the tool hooks into the well-known WEKA toolkit and uses the classifiers found therein. For more information on the WEKA toolkit see [IHW05]. The name "MAW" was chosen for this project, the name stands for *MAW And WEKA*.

The goal was to create a tool that could be easily adapted to read in different corpora. It should also be easy to add more feature extractors. It should also be easy to iterate over different configurations of the used features, as well as multiple values for some other parameters such as training-size. It was decided that it would be nice if some basic statistics where also computed and the results plotted. In order to determine how to produce the desired tool a commented concept configuration file was written;

```
//general note: comma seperation for multiple items.
[general]
corpus=...
corpustype=... // Should be a valid project.maw.readers.CorpusReader
corpusargs=... // Arguments for the Corpus (basicly: what features to read in) (:seperated)
outfile=... // Outputs the results and confusion matrixes here
plot=... // Yes or No NOTE: if plot=yes then all plotoptions are mandatory! plus linelables and restvals
// must have the same numberof different items as toplot
plottoplot=trainingsize:1,2,3,4;F0,F1,F2,F3:1,2,3,4 // the thing to plot and all its values, seperated by
// ; for multiple lines
plotrestvals=F0,F1,F2:1|F3:2;trainingsize:1 // the values of the other things that ARE ITERATED OVER
// ; for multiple lines
plottitle=... // title of the plot
plotlinelabels=line1;line2 // labels of the lines
plotxlbl=... // X-axis label
plotylbl=... // Y-axis label
plotfile=... // Save the plot somewhere
confidence=0.95 // the desired confidence interval for the delta values
usesparseinstances= // Yes or No, wether to use sparseinstances, only usefull if you think it might be ;P
// (read api on weka.core.SparseInstance and use common sense) For a small number of features "no" is recommended
[features]
F0=... // A valid project.maw.features.Feature
F0_args=.... // arguments (for example what feature to train on)
F0_name=.... // the "name" of the feature
F1=...
F1_args=...
F1_name=...
[training]
testsize=... (the number of corpusitems to use for training)
tolearn=F1 // which feature to learn
learner=...//which weka classifier to use
learnerargs=...//arguments for the classifier SPACE SEPERATED !(different then elsewhere)
[iteration]
iterations=10//number of normal iterations
trainsizes=...,...,...//comma seperated list of training-sizes (in CorpusItem’s)
F0=...,...,... //multiple iter arguments for F0
F2,F3,F4=...,...,... //iterate over arguments and give the current one to F2, F3 and F4.
```

Note that the use of the word “feature” in the configuration file can point to several things. There are corpusfeatures i.e. raw features already encoded directly in the corpus such as the text of an utterance and annotations, there are feature classes i.e. features implementing project.maw.features.Feature that generate proper features, and finally there are proper features i.e. features as they are actually used in learning.

From this concept configuration file it was determined that the feature-generators, the corpus-type, the used classifier and the iteration had to be easily changeable. A basic package structure was created with interfaces for the creation of feature generators and corpus readers. It must be noted that the feature generators must ‘fit’ the data as read in by the corpusreader. The basic structure of the final program can be seen in figure 3.4.

After the creation of the basic framework two corpusreaders and several feature generators where created. As far as the corpus readers are concerned, one was build to read in the recipe corpus and one to read in the AMI corpus. For the features first a generic nominal feature was created and a “number of words” feature. The generic nominal feature just repeats the contents of a specific corpus feature, for example what the dialogue act of an utterance is. The “number of words” feature simply counts the number of words in an corpus feature (words are sequences of one or more characters seperated by a space).

More elaborate are the set of features implementing the different cue-phrase selection methods outlines above. For this a counter class was created that simply counts phrases and targets that can be used to calculate simple statistics on these (such as $P(d|p)$). A generic cue-phrase selection feature generator was created that does all the work a feature should do except the formula that calculates the ranking of the different cue-phrases. These ranking formulas where all implemented in seperate classes that extended this cue-phrase selection

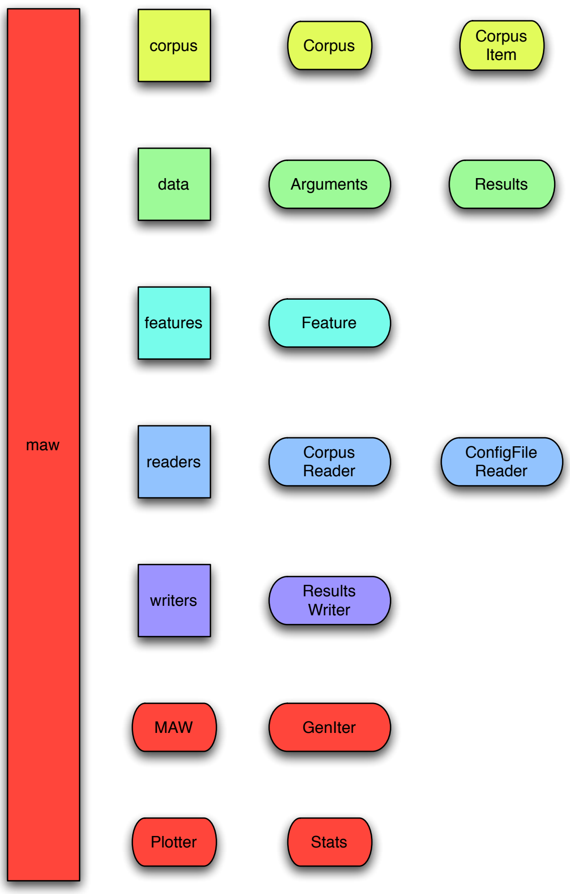


Figure 3.4: *The structure of the maw program, only lists the most important classes in each package. Packages are denoted with square containers, classes are denoted with rounded containers.*

feature generator. This set of cue-phrase selection feature generators support the selecting of x n-gram cue-phrases each iteration, meaning that each iteration can have a different number of features generated where each generated feature is the number of times the selected cue-phrase at that position occurs in the utterance.

Finally to make the tool usable for others a small user-guide was written. This user-guide is included as an appendix and it can be found in appendix A. It includes a more thorough description of the program structure, the interfaces between the different parts of the program and a short guide on adding ones own corpus formats and features to the tool. It can be concluded that the tool is very well suited for the job it was written for, and we where able to perform many experiments easily after writing the required corpusreaders and feature extractors.

CHAPTER 4

Recipes

4.1 The Recipe Corpus

In [TA06] Alofs and Latour attempted to create an automated sentence classifier for QA systems. This classifier would work in the recipe, as in recipes for cooking, domain and classify the sentences in different types. For this they used a number of features and a loglinear maximum entropy classifier. In [Rat97] Maximum Entropy models are explained clearly and in detail.

Alofs and Latour didn't yet have a corpus for this domain. They developed a system of sentence classification specific to the recipe domain and annotated the corpus with these sentence types. They hand-annotated all 3100 sentences of the recipes. Using their annotation guide and sentence classification they managed to achieve a Cohen's Kappa of 0.88 on this task. This means that they achieved a very high inter-annotator agreement on this task.

The final categories on which Alofs and Latour decided are as follows:

Requirement A requirement is something needed for cooking the dish such as an ingredient or a utensil used for preparing the dish. They note that these are usually listed at the beginning of a recipe or separate from the bulk of the recipe text.

Instruction Blocking "An instruction is a description of an action required to prepare the dish." An instruction blocking is an instruction where, during the time that the described task takes, the attention of the cook is required. This includes for instance chopping up vegetables or whipping cream.

Instruction Non-Blocking An instruction non-blocking is an instruction where, during the time that the described task takes, the attention of the cook is only required sporadically. For example "bring the water to a boil".

Emphasis The emphasis category is for sentences that direct the users attention to certain parts of the cooking process. These sentences can be used

Category	# of sentences	mean length	95% confidence δ
name	136	3.04	0.374
ins-block	903	11.21	0.361
suggestion	73	9.01	1.021
ins-nonblock	426	12.45	0.545
emphasis	30	9.57	1.946
requirement	1072	4.08	0.217
inf-irrelevant	38	6.76	1.847
signal	296	1.22	0.083
inf-relevant	80	4.84	0.517
explanation	13	12.77	4.129
other	33	8	2.441

Table 4.1: *A few simple statistics on the recipe corpus. This table contains per sentence category the number of sentences in that category, the avarage sentence length and the 95% confidence interval δ (delta).*

to help spot problem or determine the time a step in the cooking process takes. For example "the pizza is done when the cheese has melted".

Suggestion According to Alofs and Latour a suggestion is a sentence that describes an action that could add a twist to the disc or that could change the dish more to the cooks liking. For example "this dish goes well with a fresh salad".

Explanation Alofs and Latour specify that an explanation *"explains the reasonign behind an instruction, suggestion, requirement or emphasis"*. For example "this prevents the dish from getting burned.".

Information Relevant to Cooking The information relevant to cooking class contains the sentences in the recipe that contain information relevant to the cooking process such as preparation time, number of plates and nutritional information. For example "283 Kcal per person".

Name The name of the dish.

Information Irrelevant to Cooking Information about the recipe that isn't relevant to cooking. For instance the name of the author, origin of the dish and a description of the dish. For example "this recipe is typical for the south-taiwanese kitchen".

Signal Signals indicate the structure of the recipe. For example "ingredients", "preparation" and the like.

Other The other class is the bucket class, it contains everything that does not fit in any of the other categories. For example "Bon appetite".

A few short statistics about the corpus are included in table 4.1.

4.2 The Recipe Task, and Experiments

For the learning task Alofs and Latour used a mix of features. They used the complete sentence as a feature. They used a list of all unigrams, bigrams and trigrams of tokens in a sentence as feature. Also the number of words in a sentence was used because they reason it should be a good indication of certain classes such as requirement, name, signal and inf-relevant. These types of sentences are typically smaller than other sentences. They also included unigrams, bigrams and trigrams of Part-Of-Speech tags. For this they used the Medium-sized CGN tagset as they had access to a tagger that tags Dutch sentences with these tags. For more information on the CGN tagset see [vE04].

Using this combinations of these features they managed to score results of 88.2% when using the wordcount, word unigrams, bigrams and trigrams and Part-of-Speech tag unigrams. When just using the words features they managed to get results of up to 83.7%. When merging the blocking and non-blocking instruction classes the performance of the classifier could be boosted up to 93%. s Using the recipe classification task as outlined above a series of experiments where performed. For these experiments we wanted to look at the suitability of using automatically selected cue-phrases as features for the classifier. As Alofs and Latour where quite successfull in [TA06] when using all possible cue-phrases, up to length 3, as features for the Maximum Entropy learner this task should be very well suited to cue-phrase selection. We should see large gains in the first few hundred selected cue-phrases and smaller gains later on, indicating that the most indicative phrases have been selected first. In order to test this the experiments as described in [SCVS99] where recreated, only this time for the recipe corpus and the recipe sentence classification task. We can then compare the results obtained with the results of Alofs and Latour and also with the results that will be achieved on the addressee identification task (discussed in chapter 5. The experiments that we performed where:

1. For all of the automatic cue-phrase selection methods as outlined in chapter 3 we ran the following experiment. We selected different amounts of cue-phrases, specifically 0, 200, 400, 600, 800, 1000, 1200, 1400, 2000, 1600 and 3200 cue-phrases. These selected cue-phrases where then used as features on all sentences in the training set. For these experiments the J48 classifier as supplied in the WEKA toolkit was used. Verbree, in [Ver06], found this learner very effective. The experiments where run multiple times.

   This set of experiments was run to look at the performance of the different cue-phrase selection methods compared to each-other and to see if the recipe classification task is learnable using cue-phrase selection.

2. The same series of experiments was run, only this time with the lexical filtering turned on. The lexical filtering doesn't select a cue-phrase if one of the already selected cue-phrases is a subset of the cue-phrase and if the already selected cue-phrase also is selected for the same category. An example MAW configuration file can be found in figure 4.1. This configuration file shows the configuration for one of these experiments, in this particular case using the TTEST cue-phrase selection method. The experiments where run multiple times.

This second set of experiments was run to look at the effect of lexical filtering, compared to the first set of experiments.

3. Later on we also ran an experiment, using the DCP cue-phrase selection method and the J48 learner. For the number of cue-phrases we again used 0, 200, 400, 600, 800, 1000, 1200, 1400, 2000, 1600 and 3200 cue-phrases. This time we added one extra feature, namely the number of words in a sentence. For this feature a word is defined as a sequence of one or more characters separated by whitespace. The experiments where run multiple times.

   This experiment was run to see if the addition of the number of words would have an impact.

4. A final, small, experiment was run that tested the usefulness of the NaïveBayes classifier from the WEKA toolkit. DCP was again used as the cue-phrase selection method. Once again 0, 200, 400, 600, 800, 1000, 1200, 1400, 2000, 1600 and 3200 cue-phrases where selected. This time also with the number of words of the sentence as feature. This experiment was also run multiple times.

   This final set of experiments was run to see if the usage of another classifier, in this case NaïveBayes (as also used by Jovanovic in [NJN06]), has an impact on performance.

With this small series of experiments we hope to show that the cue-phrase selection methods as proposed in [SCVS99] generate a suitable set of features for a generic classification task (where one can suffice with lexical features). The final two experiments have as purpose to evaluate the suitability of the NaïveBayes classifier, as this was used extensively by Jovanic in [NJN06] (more on this in chapter 5). The results of these experiments are presented and discussed below.

4.3 Results

In this section the results for the recipe task experiments will be discussed. Each experiment, as outlined above, will be discussed separately.

As these are two terms we use in this discussion we will now define Precision and Recall. Precision and recall are two, commonly used, measurements of performance. The used definition of precision is, for classification T, the number of sentences correctly tagged with T divided by the total numver of sentences tagged with T. This is expressed in the following formula:

$$\text{precision(T)} = \frac{\#\text{ correct with T}}{\#\text{ tagged with T}}$$

The used definition of recall is, for classification T, the number of sentences correctly tagged with T divided by the number of sentences that should be tagged with T. This is expressed in the following formula:

$$\text{recall(T)} = \frac{\#\text{ correct with T}}{\#\text{ with tag T in the testset}}$$

```
[general]
corpus=/Users/qadn/Desktop/Master_thesis/Recepten-Corpus.txt
corpustype=project.maw.readers.RecipeCorpusReader
corpusargs=type:sent:pos
outfile=/Users/qadn/Desktop/recipe-TTEST.txt
plot=yes
confidence=0.95
confusionmatrix=yes
plottoplot=F0:0,200,400,600,800,1000,1200,1400,2000,2600,3200
plotrestvals=trainingsize:2500
plottitle=TTEST
plotlinelabels=TTEST
plotxlbl=Number of Phrases
plotylbl=Accuracy
plotfile=/Users/qadn/Desktop/recipe-TTEST.eps
usesparceinstances=yes
[features]
F0=project.maw.features.PhraseTTESTFeature
F0_args=sent:type:3:yes
F0_name=ENT
F1=project.maw.features.GenericNominalFeature
F1_args=type
F1_name=type
[training]
testsize=600
tolearn=F1
learner=weka.classifiers.trees.J48
learnerargs=
[iteration]
iterations=10
trainsizes=2500
F0=0,200,400,600,800,1000,1200,1400,2000,2600,3200
```

Figure 4.1: *An example configuration file for the recipe classification task. This particular configuration file constructs an experiment using the TTEST cue-phrase selection method for selecting the cue-phrases with a trainingsize of 2500 sentences and a testingsize of 600 sentences. Lexical filtering is used.*

In short, precision is a measure of the accuracy of a classification, and recall is a measure of the completeness of a classification. If a certain class has a high precision rating, but a low recall then too few utterances are labeled with it. On the other hand, a low precision and high recall means that too many utterances are labeled with it.

1. The first set of experiments as run on the recipe classification task uses all different cue-phrase selection methods, as introduced in [SCVS99] and discussed in chapter 3, on the recipe corpus. With each of these methods 0, 200, 400, 600, 800, 1000, 1200, 1400, 2000, 2600 and 3200 cue-phrases where selected to be used as features. The learning was done using the J48 classifier from the weka toolkit using MAW. The results of these experiments can be found in table 4.2.

   Most cue-phrase selection methods get good results already when selecting only 200 cue-phrases, DCP lags a bit behind with good results from 400 cue-phrases. In [SCVS99] DCP was also observed to lag behind in the beginning. The exception to this are the ENT, S and TTEST methods. These also performed quite badly in [SCVS99]. The performance achieved, as seen from the results, varies 76% and 82%. It is noticeable that the achieved performance of the classification seems to sometimes peak early and then go down slightly again, such as in the results of, amongst others, COOC. It should be noted however that the results reported in [SCVS99] show the same small valleys.

   Strangely enough the D (or **deviation**) cue-phrase selection method seems to perform quite well. In [SCVS99] this was one of the worst performers.

   Overall the results achieved in these experiments are in line with those reported in [TA06]. Their results, without adding Part-of-Speech n-grams as features[1] , are slightly better.

We see that the COOC,CP,D, DCP, IG and MI methods show the desired behavior: fast increase for a small number of cue-phrases, with the increase tapering of for larger number of cue-phrases.

2. For the second set of experiments we performed the same experiment as above, but with a small difference. This time the lexical filtering, as proposed in [SCVS99], was used. As explained before in chapter 3 the lexical filtering removes possible cue-phrases of which a sub-ngram has already been selected for the same target. The results of this set of experiments can be found in table 4.3.

   Most cue-phrase selection methods get good results at 200 selected cue-phrases. again with DCP lagging. Just like the previous experiment we see the TTEST, S and ENT methods not performing as well as the rest. Once again we observe the small valleys in the results.

   When comparing these results with the results of the previous experiment (as they are listed in table 4.2) we notice that overall the performance is better with most of the methods (except TTEST, S and ENT) achieving an 80% or better accuracy. Especially the COOC, CP and IG methods benefit from the lexical filtering. However the results of the MI cue-phrase selection method are worse with filtering than without filtering, in [SCVS99] lexical filtering improved its results.

   Overall the results achieved in these experiments are in line with those reported in [TA06]. Their results, without adding Part-of-Speech n-grams as features, are slightly better.

3. For the third set of experiments on the Recipe corpus we used the DCP cue-phrase selection method to select 0, 200, 400, 600, 800, 1000, 1200, 1400, 2000, 2600 and 3200 cue-phrases. The J48 learner was used again and the experiment was run several times. This small set of experiments will be reported in a bit more detail.

   A plot of the results can be found in figure 4.2, with a corresponding confusion matrix of one of the runs in table 4.4. This confusion matrix was taken from a single run in which 3200 cue-phrases where selected as features. Of the same run an recall-precision table is included as well in table 4.5.

   From the plot we can see that using the DCP selection method shows the desired behavior, with the first n cue-phrases delivering the biggest increase in accuracy.

   The same set of experiments was run, but this time with the addition of the number of words in a sentence as feature. The results of these can be found in figure 4.3 for a plot of the results, in table 4.6 for the confusionmatrix of a single run when selecting 3200 cue-phrases and in table 4.7 for the recall and precision table of that same run.

[1] They also produced even better results when using part-of-speech n-grams, however we do not compare with those results as it would skew the comparison. We don't use part-of-speech n-grams after all.

Looking at the results we see that the addition of the number of words increases the baseline accuracy of the classifier. The DCP method again shows the desired behavior here, only less obvious as before. There is less accuracy to be gained with a higher base-accuracy of the classifier.

Looking at the two confusion matrixes we see that the major misclassifications are sentences misclassified as instruction blocking and requirement. It is interesting to note that these two categories are also the two most misclassified. The recall and precision tables show high recall and precision ratings for requirement, signal, ins-block and ins-nonblock in both experiments. The biggest difference can be seen in the other and inf-relevant categories. In the first experiment a higher number of inf-relevant seems to have been misclassified as signal. However, considering the low total number of inf-relefant sentences It is hard to draw conclusions from this information. The big difference in the classification of the other category seems to stem from a misclassification of a sentence from the emphasis category in the second experiment, which halved the precision due to the extremely low number of sentences of type "other" in the test-set and the corpus. Considering the small number of sentences involved it is hard to draw conclusions. The vast majority of misclassifications and misclassified sentences are of the requirement or ins-block types as these types make up the majority of the corpus.

The differences between not using the number of words in a sentence and using the number of words in a sentence is fairly small. With only the initial performance, when using very small numbers of cue-phrases, differing. Though, when using only 200 cue-phrases as features, the performance with the number of words features was 10% higher, around 70%.

4. Finally we look at the results with NaïveBayes as they where achieved on the recipe classification task. On the recipe corpus we performed the same small set of experiments twice. The second time we added the number of words in a sentence as feature.

   The results for the first of these two sets of experiments can be found in figure 4.4. As we can see from this plot the results start out at the well-known majority class classification at an accuracy of around 36%. However then the achieved performance plummets for the first 200 selected cue-phrases. From then on the performance achieved slowly increases until it it reaches around 49% when selecting 3200 cue-phrases to use as features for learning.

   The results of the second of these two sets of experiments can be found in figure 4.5. As one can see from this plot, when one only knows the number of words in the sentence as a feature the NaïveBayes classifier manages to achieve a performance of around 56%. However, when selected cue-phrases are added as features the performance plummets sharply to around the same point as in the first experiment. Eventually, when adding more and more cue-phrases a performance of 48% is reached.

   It is clear that, in combination with a NaïveBayes learner, DCP does not show the desired early peak in performance. The results as a whole, even when at 3200 selected cue-phrases is bad when compared to the results when using the J48 learner. We can therefore conclude that the choice of

# cue-phrases	COOC	CP	D	DCP	ENT	IG	MI	S	TTEST
0	0.343	0.365	0.338	0.338	0.347	0.355	0.368	0.313	0.327
200	0.737	0.747	0.742	0.685	0.347	0.753	0.757	0.315	0.328
400	0.742	0.753	0.767	0.752	0.347	0.728	0.787	0.348	0.347
600	0.758	0.775	0.795	0.748	0.348	0.760	0.787	0.348	0.347
800	0.778	0.765	0.773	0.753	0.348	0.768	0.787	0.435	0.347
1000	0.778	0.768	0.773	0.747	0.347	0.773	0.805	0.438	0.347
1200	0.757	0.763	0.768	0.750	0.350	0.773	0.803	0.438	0.347
1400	0.772	0.760	0.768	0.760	0.397	0.768	0.820	0.438	0.453
2000	0.790	0.747	0.762	0.782	0.627	0.775	0.795	0.442	0.453
2600	0.775	0.763	0.765	0.773	0.652	0.777	0.795	0.642	0.453
3200	0.782	0.768	0.768	0.758	0.658	0.778	0.822	0.657	0.447

Table 4.2: *A table containing the results of the experiments with different cue-phrase selection methods and different numbers of cue-phrases on the recipe classification task. This is done on the recipe corpus and without the lexical filter.*

# cue-phrases	COOC	CP	D	DCP	ENT	IG	MI	S	TTEST
0	0.362	0.372	0.353	0.338	0.387	0.383	0.350	0.360	0.358
200	0.735	0.778	0.737	0.720	0.328	0.742	0.757	0.393	0.358
400	0.742	0.795	0.762	0.745	0.328	0.758	0.748	0.405	0.385
600	0.755	0.783	0.778	0.745	0.337	0.742	0.753	0.493	0.385
800	0.770	0.778	0.785	0.725	0.338	0.743	0.753	0.498	0.387
1000	0.780	0.785	0.780	0.728	0.338	0.758	0.772	0.507	0.387
1200	0.775	0.788	0.785	0.757	0.338	0.750	0.763	0.518	0.485
1400	0.793	0.775	0.777	0.750	0.338	0.755	0.765	0.640	0.485
2000	0.802	0.808	0.798	0.772	0.628	0.797	0.790	0.692	0.488
2600	0.803	0.798	0.798	0.793	0.642	0.798	0.797	0.750	0.490
3200	0.808	0.805	0.798	0.795	0.647	0.812	0.783	0.752	0.490

Table 4.3: *A table containing the results of the experiments with different cue-phrase selection methods and different numbers of cue-phrases on the recipe classification task. This is done on the recipe corpus and with the lexical filter.*

classifier is an important one, the classifier must me suitable to be used with cue-phrase selection methods.

Generally the COOC, CP, D, DCP, IG and MI methods display the desired behavior, with the first n selected cue-phrases contributing significantly more to the achieved performance than the later ones.

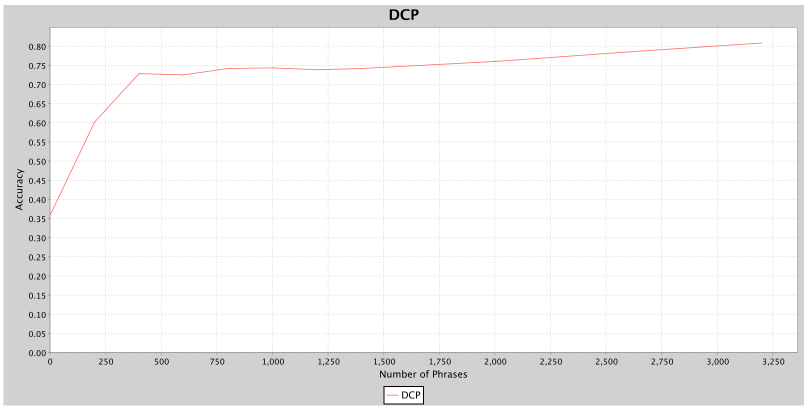


Figure 4.2: *Results of lexically filtered cue-phrase selection using the DCP method on the recipe classification task, using J48 as learner. This plot provides a clear example of the desired behavior of cue-phrase selection.*

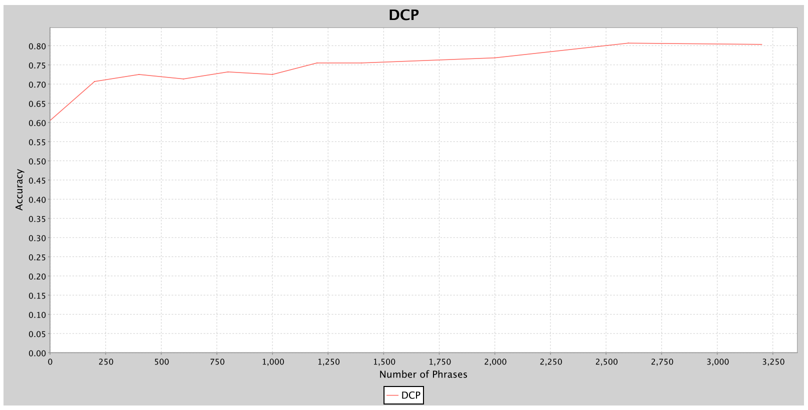


Figure 4.3: *Results of lexically filtered cue-phrase selection using the DCP method on the recipe classification task, using J48 as learner. With as extra feature the length of the utterance.*

	0	1	2	3	4	5	6	7	8	9	10
0:requirement	198	1	5	0	0	2	0	3	4	0	2
1:signal	6	54	0	0	0	0	0	0	0	0	0
2:ins-block	8	0	157	1	11	1	0	4	1	0	0
3:emphasis	0	0	3	1	0	1	0	1	0	0	0
4:ins-nonblock	2	0	21	0	54	0	0	0	0	0	0
5:suggestion	1	0	3	0	0	6	0	0	0	0	0
6:explanation	0	0	2	0	0	0	0	0	0	0	0
7:inf-irrelevant	1	1	1	0	0	1	1	1	0	0	0
8:name	7	0	3	0	0	1	0	0	6	0	0
9:other	1	0	4	0	0	2	0	0	1	1	0
10:inf-relevant	2	4	1	0	1	0	0	0	0	0	7

Table 4.4: *Confusion Matrix for an experiment with 3200 DCP selected cue-phrases on the recipe classification task, using J48 as learner on the recipe corpus. 0: requirement 1: signal 2: ins-block 3: emphasis 4: ins-nonblock 5: suggestion 6: explanation 7: inf-irrelevant 8: name 9: other 10: inf-relevant*

Item	Recall	Precision
requirement	0.92	0.88
signal	0.9	0.9
ins-block	0.86	0.79
emphasis	0.17	0.5
ins-nonblock	0.70	0.82
suggestion	0.6	0.43
explanation	0.0	0.0
inf-irrelevant	0.17	0.11
name	0.35	0.5
other	0.11	1.0
inf-relevant	0.47	0.78

Table 4.5: *Recall and Precision for an experiment with 3200 DCP selected cue-phrases on the recipe classification task, using J48 as learner on the recipe corpus.*

	0	1	2	3	4	5	6	7	8	9	10
0:requirement	211	0	7	0	1	1	0	0	1	0	1
1:signal	2	55	1	0	0	0	0	0	0	0	1
2:ins-block	5	0	136	0	17	2	0	0	0	0	0
3:emphasis	0	0	4	0	0	0	0	0	0	1	0
4:ins-nonblock	3	0	19	0	61	0	0	0	0	0	0
5:suggestion	1	0	10	0	0	4	0	0	0	0	1
6:explanation	1	0	1	0	1	0	0	0	0	0	0
7:inf-irrelevant	3	0	4	0	0	0	0	1	0	0	1
8:name	17	0	3	1	0	0	0	0	3	0	0
9:other	0	0	3	1	0	0	0	0	0	1	0
10:inf-relevant	2	1	1	0	0	0	0	0	0	0	10

Table 4.6: *Confusion Matrix for an experiment with 3200 DCP selected cue-phrases and the number of words in an utterance on the recipe classification task, using J48 as classifier. 0: requirement 1: signal 2: ins-block 3: emphasis 4: ins-nonblock 5: suggestion 6: explanation 7: inf-irrelevant 8: name 9: other 10: inf-relevant*

Item	Recall	Precision
requirement	0.95	0.86
signal	0.93	0.98
ins-block	0.85	0.72
emphasis	0.0	0.0
ins-nonblock	0.73	0.76
suggestion	0.25	0.57
explanation	0.0	ERR
inf-irrelevant	0.11	1.0
name	0.13	0.75
other	0.2	0.5
inf-relevant	0.71	0.71

Table 4.7: *Recall and Precision for an experiment with 3200 DCP selected cue-phrases and the number of words in an utterance, using J48 as learner on the recipe classification task.*

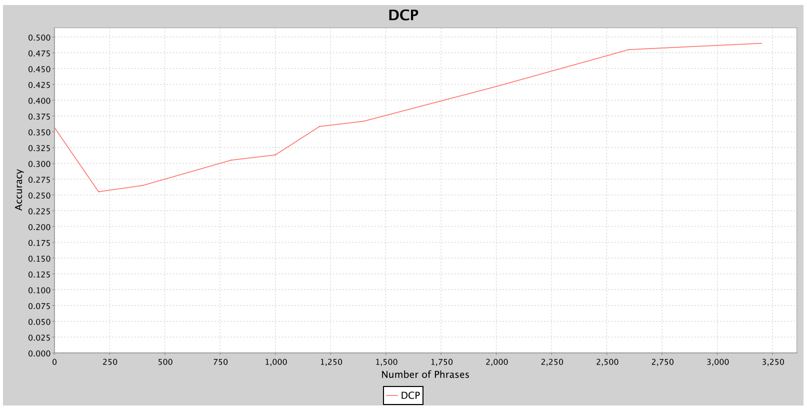


Figure 4.4: *Results of lexically filtered cue-phrase selection using the DCP method on the recipe classification task, using NaiveBayes as learner.*

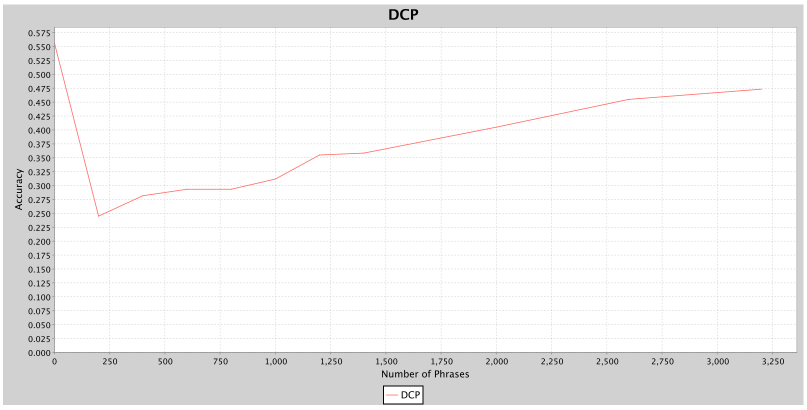


Figure 4.5: *Results of lexically filtered cue-phrase selection using the DCP method on the recipe classification task, using NaiveBayes as learner. With as extra feature the length of the utterance.*

CHAPTER 5

Addressing

Addressing is an important part of the conversational flow of meetings. By addressing a (sub-)group of the meeting participants the flow of the conversation is steered. It is steered in such a way that it is clear to the participants whom the speaker expects to respond. The person or persons that are expected to respond are denoted as the addressee of that utterance. In [Gof81] Goffman defines the addressee(s) as: *"ratified participants () oriented to by the speaker in a manner to suggest that his words are particularly for them, and that some answer is therefore anticipated from them, more so than from the other ratified participants"*. Goffman (in [Gof81]) defines several types of participants, including ratified and non-ratified participants. Ratified participants are those participants to a conversation that are included in the normal conversational flow. For example, someone that is listening in on a conversation from a, perhaps small, distance is a non-ratified participant. Since we concern ourselves with meetings this distinction is irrelevant; All participants are ratified participants. With the addressee, be it single or group addressed we simply mean the ratified participant(s) that the speaker anticipates/wants to respond.

In section 5.1 we will look at the work done on the detection of addressee in meetings by Jovanovic. In this chapter we will introduce the addressee task; We will look at the learn-ability of addressee in meetings, specifically if it is possible to learn whether a phrase is single or group addressed. This will be done using the various cue-phrase selection methods described in chapter 3. In short we will try to answer the question: *"Can we determine if a phrase is single or group addressed based on automatically selected cue-phrases?"*, and this chapter provides the introduction to that task, and the setup of the experiments that where done will be discussed.

5.1 Results of Jovanovic

According to Goffman [Gof81] there are three basic kinds of hearers; Those who *overhear*; Those who are ratifed, but are not *specifically* addressed by the speaker (also called *unaddressed*); And those ratified participants who are *addressed.* Jovanovic focusses in [NJN06] only on the ratified participants. *"Therefore, the problem of addressee identification amounts to the problem of distinguishing addressed from unaddressed participants* **for each dialogue act** *that speakers perform."*[NJN06].

To learn this a Bayesian Networks and a Naïve Bayes learner where used as well as a rather large feature set:

Contextual Features Several contextual features where used that provide information about the preceding utterances. Information about the speaker, the dialogue act and the addressee of the previous utterance where used. In addition to this the same information was included about the related utterance. Jovanovic defines this related utterance as: *"A related utterance is the utterance that is the a-part of an adjacency pair wih the current utterance as the b-part."*[NJN06]. She also included information about the speaker of the current utterance.

Utterance Features As utterance features several simple utterance features where used: Whether the utterance contains personal pronouns? Whether the utterance contains possessive pronouns or adjectives? Whether the utterance contains indefinite prodouns? Whether the utterance contains the name of participant P_x? What the dialogue act of the utterance is. And finally whether the utterance is short, an utterance is considered short if its duration is less or equal to one second.[NJN06].

Gaze Features A set of Gaze features was included for each Participant P_x, namely P_x-looks-P_y and P_x-looks-NT where $x, y \in [0, 1, 2, 3]$ and $x \neq y$. Here P_x-looks-NT represents that participant P_x doesn't look at any of the participants. The value of these features represents the number of times that the participants look at another participant or away during the single utterance. The values of these features is *zero* for 0, *one* for 1, *two* for 2 and *more* for three or more times. In the second experiment this information was only included about who the speaker of the utterance looked at.

Meeting Context As meeting context feature a single feature was used that represents the meeting actions, in the first experiment they used the full set of speech based meeting actions[1]. For the second experiment consensus, disagreement and the discussion meeting actions where grouped under a single category marked discussion.

For these experiments Jovanovic used the M4 corpus which is annotated with a Dialogue act set based on the MRDA set. The annotations of the M4 corpus, such as the segmentation and the dialogue acts, have a very high inter-annotator agreement. The inter-annotator agreement of the dialogue act annotation, the addressee annotation and the Gaze information have very high kappa values (all with $\kappa \geq 0.70$). About 40% of the utterance is addressed to all participants and the rest is distributed relatively evenly across the different participants as one can see in table 5.1.

[1]monologue, discussion, presentation, white-board, consensus and disagreement

ALLP	P_0	P_1	P_2	P_3
40.20%	13.83%	17.03%	15.88%	13.06%

Table 5.1: *The distribution of addressee values in the M4 corpus. Taken from [NJN06]*

For the first set of experiments the results with all features are an accuracy of 81.69% for the Bayesian Network and 78.23% fro the Naïve Bayes classifier. For the second set of experiments those results are 83.74% and 79.13% respectively. Note that only the results for the 4 participant meetings are listed here since they are the most relevant to the results of the AMI corpus. Jovanovic also performed a similar set of experiments on the AMI corpus, when using exactly the same feature set as for the M4 corpus an accuracy of 76.77% was reached [NJN06]. With some tweaking of the feature set this was eventually raised to 78.37% when using a Bayesian Network and 77.56% when using NaïveBayes. The tweaking was done by including a few new features, namely whether or not the sentence is reflexive and adding certain context information (is the user drawing, etc), also the length-of-utterance feature was quantified to {one, few, more (> 4)}. From the confusion matrixes in [NJN06] it becomes clear that the system confuses P_x much more with ALLP then with P_y.

5.2 Addressee and Discussions

A statistic which one expects to find in discussions is that of certain addressee patterns. Specifically it is expected that during a discussion the addressing and speaker turns jump back and forth. Meaning that after participant A addresses participant B one expects B to respond to A, and back and forth. This is expected to be more pronounced during discussions than outside of discussions.

So, using the available annotations of discussions and addressing[2], the distribution of speaker-addressee patterns for adjacency pairs was counted. These numbers where generated for all adjacency pairs inside and all adjacency pairs outside discussions. The results, seperated by discussion, annotator and inside/outside discussions, can be found in appendix D. Though the expected pattern (ABBA) is found more often in discussions for annotator H, the difference is not visible for the other annotators. Unfortunatly, as the definition used for the annotation of discussions was insufficient, this result can only be seen as an indication that our expectation was false.

5.3 The AMI Corpus

The AMI corpus consists of recordings and transcriptions of a large number of meetings. Various annotations are applied to the meetings, including dialogue acts, addressee information, gaze information and participant gestures and postures.

The meetings in the AMI corpus are four participant meetings, where four people roleplay the four roles based on a scenario. The four roles that are used in

[2]These two annotations where both available for meetings IS1003c and IS1003d only.

these meetings are that of the project manager, marketing expert, user interface designer and industrial designer. Meetings are recorded in a set of four meetings, in which the four participants design an innovative TV remote control.

5.4 Method

In order to answer the question *"Can we determine if a phrase is single or group addressed based on automatically selected cue-phrases?"* several experiments where set up. For these experiments a part of the AMI corpus was available that was already annotated with the addressee information (from the work of Jovanovic).

The J48 learner used in most of the experiments is an implementation of the C4.5 tree classification algorithm and is part of the WEKA toolkit[IHW05].

The experiments where done using the MAW tool, which is described shortly is section 3.3 and in detail in appendix A. For this tool a corpusreader was made for the addressing data. All the necessary feature selectors where implemented, which was a trivial since the complex feature selectors for the cue-phrases where already implemented.

In these experiments several datasets where used, all based on all available addressee annotated data. The first containing the entire addressee annotated part of the AMI corpus contains 3372 group addressed sentences, 2754 single addressed sentences and 3861 unknown addressed sentences. The second version of the corpus, referred to as the filtered corpus, had all utterances that where annotated with a backchannel, stall, fragment or other dialogue act removed, as these have per definition no addressee, this leaves 262 unknown sentences. A final version of the corpus was also used briefly that simply had all sentences with the addressee annotation "unknown" removed.

Using this tool and the AMI corpus several experiments where performed.

1. The first set of experiments as run on the addressee task uses all different cue-phrase selection methods, as introduced in [SCVS99] and discussed in chapter 3. With each of these cue-phrase selection methods 0,200,400,600,800,100,1200,1400,2000,2600 and 3200 cue-phrases where selected to be used as features. J48 was used as the classifier. The experiment was run on the first, "raw", version of our trainingset.

   This series of experiments was run to see how the cue-phrase selection methods perform compared to each-other and to see if the single/group addressed classification task is learnable using cue-phrases.

2. The second series of experiments that was run was almost identical to the first. The only small difference is that this time the lexical filtering, as described in [SCVS99], was turned on.

   This experiment was run to look at the effect of lexical filtering.

3. The third set of experiments once again uses all different cue-phrase selection methods to select different numbers of cue-phrases. These methods are used to select 0,200,400,600,800,100,1200,1400,2000,2600 and 3200 cue-phrases. Once again we use the J48 classifier for learning. The difference between this and the first series of experiments is that this series

is performed using the modified corpus for learning. This filtered version had utterances of certain Dialogue Act types removed.

This series of experiments was run to see what the effect of the removal of the unclassifiable utterances would have.

4. We turn once again to nearly the same experimental setup, this time almost identical to the third series of experiments. The only difference with the previous series being that lexical filtering has been used on the selected cue-phrases.

   This experiment was run to look at the effect of lexical filtering.

5. When we looked at the skewed selection of cue-phrases, as van be seen in appendix F.1, it became apparent that a disproportional number of them where for the few sentences annotated with unknown in the filtered corpus. Therefore a small experiment using DCP with lexical filtering on the addressee corpus with all sentences, annotated with the unknown addressee, filtered out.

   This experiment was run to see if the removal of all utterances marked "unknown" would improve results.

6. A lot of experiments where performed where we only used cue-phrases as features for training the classifier. Verbree showed in [Ver06] that the length of an utterance is an important feature for classification, as well as the dialogue act of an utterance. Therefore we performed a series of experiments that, next to the selected cue-phrases, uses these two features for learning. Using once again the DCP cue-phrase selection method we selected 0, 200, 400, 600, 800, 1000, 1200, 1400, 2000, 1600 and 3200 cue-phrases as features and added the length of an utterance and the dialogue act that utterance is annotated with as features.

   This experiment was run to see what sort of effect the addition of the number of words in an utterance and the dialogue act has.

7. We ran a two experiments with the LIT set of cue-phrases as features. Both [SCVS99] and [Kat06] used a LIT set for comparison, but we happened to have easy access to the LIT set as used in [Kat06], so that one was used. This experiment was done on the filtered version of the AMI corpus. The second experiment of the two included the number of words of the utterance and the Dialogue Act of the utterance as features. For both experiments the J48 classifier was used.

   These two experiments where performed to see how the LIT set would do.

8. In [NJN06] Jovanovic used Bayes based classifiers, including a NaïveBayes classifier. With these she achieved good results. Using the DCP cue-phrase selection method we ran a small series of experiments, selecting 0, 200, 400, 600, 800, 1000, 1200, 1400, 2000, 1600 and 3200 cue-phrases as features. These experiments where done on the filtered version of the AMI corpus and with lexical filtering turned on. For this we used the NaïveBayes classifier instead of the J48 classifier.

   This final set of experiments was run to see if the usage of another classifier, in this case NaïveBayes (as also used by Jovanovic in [NJN06]), has an impact on performance.

The results of these experiments will be discussed next. The conclusions will be drawn in the conclusion chapter, which concludes this thesis.

5.5 Results of the Addressee Classification Task

1. For our first set of experiments the results are reported in table 5.2. This is the set experiments over all cue-phrase selection methods selecting 0, 200, 400, 600, 800, 1000, 1200, 1400, 2000, 2600 and 3200 cue-phrases as features on the raw AMI corpus with addressee data, without using lexical filtering in the selection of the cue-phrases.

   As shown in the table the ENT, S and TTEST methods perform particularly bad, with the ENT method catching up to the rest when using a large number of cue-p[hrases (2600 and 3200). These same methods didn't work well in [SCVS99], but unlike there the D selection method seems to work well.

   The overall best result lies around 65% and is produced with the COOC cue-phrase selection method when selecting 2000 or 2600 cue-phrases. Nevertheless these results are quite poor, especially considering the results achieved by Jovanovic.

2. The same set of experiments was also performed with lexical filtering, and the results of those can be found in table 5.3. Overall the lexical filtering helps a little, allowing the performance to peak with a smaller number of cue-phrases. However the overall performance is again poor, achieving results of around 65% at best.

   The differences between with and without lexical filtering are slight, with D and IG performing slightly worse with lexical filtering than without and CP, DCP, ENT, S and TTEST performing slightly better with lexical filtering than without. The overall best result is still 65% , which is still poor compared to the results of Jovanovic.

3. When using the filtered AMI addressee set the results didn't get much better. The filtered set is obtained by removing utterances annotated with the backchannel, stall, fragment or 'other' dialogue acts. On this corpus we performed a set of experiments using all cue-phrase selection methods to select 0, 200, 400, 600, 800, 1000, 1200, 1400, 2000, 2600 and 3200 cue-phrases as features. The results of this set of experiments can be found in table 5.4.

   Again, the overall best result achieved lies at 65%. This time it is achieved by the ENT cue-phrase selection method at 3200 cue-phrases. When not having selected any cue-phrases, so when learning without any features the results lie between 50 and 55%. This difference in performance can be explained by the fact that the training and test set are randomly selected by the tool, since when not selecting any cue-phrases it only depends on the learning algorithm used. In this case the algorithm will select the majority class in the training set to use as the classification.

   We speculate that the reason for this lack of a performance increase is due to the relative ease with which unknown sentences could be classified in the previous experiments. This theory is supported by the high number cue-phrases found that indicate unknown addressee, the list of the first 200 found cue-phrases of a single run can be found in appendix F.1.

4. Once again the same set of experiments was carried out. This time On the filtered corpus with lexical filtering turned on.w The results of these experiments can be found in table 5.5.

   The best result achieved was with the TTEST cue-phrase selection method when selecting 2600 cue-phrases. Even so this result still lies at only 65%.

   The differences between the results with and without lexical filtering are, overall, not large. The largest differences for the better can be seen for the IG and TTEST cue-phrase selection methods. The IG method mainly gains from this when selecting relatively low numbers of cue-phrases, peaking at 400. While the TTEST method gains at the end, at 2600 and 3200 cue-phrases. The largest differences for the worse can be found in the ENT and S methods. Both not getting as much accuracy when selecting large numbers of cue-phrases as they did without lexical filtering.

   For each cue-phrase selection method a single run, selecting 3200 cue-phrases as features, was used to create a confusion matrix. The confusion matrixes for all the cue-phrase selection methods can be found in table 5.6. Most of these confusion matrixes are quite symmetrical, while for most methods confusing sentences as type "unknown" is low.

   Note that the the confusion as single addressed is somewhat high in CP, D and DCP and very high in S. On the other side the confusion as group addressed is high for the TTEST method and extemely high for the ENT method.

5. A small experiment was performed on the corpus with all utterances annotated with the addressee "unknown" filtered out. For this the DCP cue-phrase selection method was used to select 3200 cue-phrases that where used as features. The J48 classifier from the WEKA toolkit was used as learner. With this we achieve an average accuracy of around 64%. A confusion matrix of one of the runs of this experiment can be found in table 5.7. in appedix F.2 a list of the 200 most important selected cue-phrases of a single run can be found.

   Not much to say about this list, except two small observations. First of all a large percentage of these cue-phrases belong to the group-addressed classification. Secondly a large percentage of these are unigrams. Quite a large number, especially at the start of the list, are not cue-phrases which we would have selected such as "the", "and" and "a".

6. Another small experiment was performed which added the dialogue act of the utterance and also its length as features in addition to the selected cue-phrases. The cue-phrases where selected using DCP and 0, 200, 400, 600, 800, 1000, 1200, 1400, 2000, 2600 and 3200 cue-phrases where selected as features. A plot of the results of this experiment can be found in figure 5.2. It is interesting to note that the results when selecting no cue-phrases as features are better than the results with cue-phrases, namely around 64%.

7. A small experiment using the LIT set from [Ver06] as features produced results of 61%. With the addition of the length and the dialogue act of an utterance this became around 63%. Again it should be noted that the results without the LIT cue-phrases as features and with only the dialogue act and the length of an utterance as features the performance is around 64%.

8. We also ran a single set of experiments on the addressee classification task, without the addition of extra features besides the cue-phrases.

   The results of this set of experiments can be found in figure 5.1. When not using any cue-phrases for learning the classifier achieves a performance of around 50%. However, as we can see the results show the same, familiar, undesirable plummet of accuracy when adding cue-phrases as features. When adding more and more cue-phrases the results show no large increase in accuracy, contrary to the results of the recipe corpus, just staying short of an 18% accuracy at the end. The reason the results don't climb slightly, as they did on the recipe task, when adding more cue-phrases is probably the fact that cue-phrases seem ill-suited to the addressee identification task.

   It is clear that, once more, the NaïveBayes learner when coupled with the cue-phrase selection method DCP provides undesirable, not providing the same early peak in performance that we saw on the same setup while using the J48 learner. The NaïveBayes classifier provides bad results compared to the J48 classifier on this task, just as it did for the recipe task. So NaïveBayes is not suitable when used in combination with cue-phrases as features.

Generally speaking the COOC, CP, D, DCP, IG and MI methods display the desired behavior, with the first cue-phrases contributing significantly more to the achieved performance than the later ones. These are the same cue-phrase selection methods that perform well as those on the recipe task. So we can conclude that these methods work well, and can be used to achieve consistent desirable results and that the ENT, S and TTEST methods should be avoided.

However the results on the classification task itself are very poor. When just using the dialogue act of a phrase and the number of words we achieve a higher accuracy than with cue-phrases as features. This is unlike the results of Jovanovic ([NJN06]) who achieved around 80% accury while predicting the exact addressee. Jovanovic did use other, more computationally expensive, features such as the gaze and gestures.

# cue-phrases	COOC	CP	D	DCP	ENT	IG	MI	S	TTEST
0	0.388	0.411	0.388	0.372	0.382	0.380	0.383	0.382	0.393
200	0.638	0.632	0.628	0.611	0.401	0.632	0.636	0.382	0.401
400	0.649	0.636	0.619	0.616	0.402	0.642	0.637	0.405	0.433
600	0.647	0.635	0.631	0.601	0.415	0.633	0.636	0.426	0.468
800	0.645	0.633	0.608	0.639	0.420	0.629	0.637	0.439	0.474
1000	0.643	0.633	0.630	0.608	0.430	0.633	0.633	0.457	0.451
1200	0.637	0.635	0.625	0.615	0.426	0.636	0.625	0.474	0.514
1400	0.635	0.625	0.632	0.607	0.453	0.641	0.627	0.456	0.492
2000	0.649	0.636	0.620	0.625	0.538	0.643	0.639	0.483	0.510
2600	0.647	0.623	0.633	0.617	0.614	0.635	0.631	0.478	0.534
3200	0.635	0.623	0.623	0.604	0.616	0.634	0.625	0.498	0.493

Table 5.2: *A table containing the results of the experiments with different cue-phrase selection methods and different numbers of cue-phrases for the addressee identification task. This is done on the unfiltered corpus and without the lexical filter.*

# cue-phrases	COOC	CP	D	DCP	ENT	IG	MI	S	TTEST
0	0.377	0.372	0.374	0.394	0.412	0.354	0.372	0.390	0.394
200	0.645	0.631	0.609	0.620	0.430	0.599	0.641	0.398	0.394
400	0.648	0.613	0.595	0.621	0.423	0.617	0.650	0.426	0.533
600	0.651	0.627	0.601	0.631	0.444	0.614	0.639	0.436	0.453
800	0.645	0.642	0.600	0.629	0.438	0.611	0.639	0.470	0.467
1000	0.649	0.619	0.612	0.611	0.446	0.605	0.627	0.433	0.516
1200	0.645	0.641	0.595	0.613	0.447	0.625	0.643	0.477	0.479
1400	0.645	0.632	0.605	0.621	0.442	0.623	0.642	0.494	0.539
2000	0.647	0.629	0.615	0.629	0.635	0.609	0.634	0.491	0.462
2600	0.645	0.617	0.595	0.623	0.559	0.607	0.631	0.508	0.456
3200	0.635	0.627	0.613	0.625	0.639	0.603	0.623	0.508	0.554

Table 5.3: *A table containing The results of the experiments with different cue-phrase selection methods and different numbers of cue-phrases for the addressee identification task. This is done on the unfiltered corpus and with the lexical filter.*

# cue-phrases	COOC	CP	D	DCP	ENT	IG	MI	S	TTEST
0	0.534	0.496	0.553	0.536	0.516	0.505	0.511	0.534	0.503
200	0.613	0.623	0.623	0.611	0.516	0.611	0.614	0.534	0.503
400	0.621	0.625	0.620	0.606	0.516	0.610	0.620	0.534	0.503
600	0.611	0.604	0.618	0.618	0.516	0.623	0.611	0.534	0.566
800	0.616	0.615	0.603	0.617	0.516	0.597	0.610	0.534	0.503
1000	0.595	0.616	0.626	0.627	0.516	0.619	0.601	0.534	0.503
1200	0.599	0.615	0.629	0.603	0.516	0.609	0.601	0.534	0.503
1400	0.605	0.629	0.621	0.625	0.516	0.625	0.593	0.584	0.503
2000	0.599	0.604	0.619	0.596	0.516	0.594	0.603	0.579	0.503
2600	0.605	0.604	0.599	0.593	0.512	0.602	0.604	0.533	0.503
3200	0.603	0.617	0.607	0.614	0.647	0.613	0.593	0.592	0.575

Table 5.4: *A table containing The results of the experiments with different cue-phrase selection methods and different numbers of cue-phrases for the addressee identification task. This is done on the filtered corpus and without the lexical filter.*

# cue-phrases	COOC	CP	D	DCP	ENT	IG	MI	S	TTEST
0	0.532	0.524	0.515	0.514	0.492	0.536	0.544	0.553	0.511
200	0.607	0.639	0.621	0.634	0.492	0.629	0.613	0.553	0.511
400	0.606	0.625	0.619	0.627	0.492	0.640	0.625	0.553	0.511
600	0.599	0.621	0.630	0.620	0.492	0.623	0.625	0.553	0.511
800	0.603	0.602	0.621	0.632	0.492	0.625	0.609	0.553	0.511
1000	0.605	0.615	0.609	0.631	0.492	0.627	0.614	0.553	0.511
1200	0.601	0.599	0.611	0.612	0.492	0.631	0.601	0.593	0.511
1400	0.605	0.616	0.617	0.587	0.492	0.617	0.623	0.534	0.511
2000	0.603	0.606	0.607	0.607	0.492	0.622	0.615	0.528	0.511
2600	0.602	0.597	0.605	0.603	0.495	0.607	0.595	0.528	0.650
3200	0.602	0.605	0.613	0.597	0.498	0.605	0.607	0.548	0.647

Table 5.5: *A table containing The results of the experiments with different cue-phrase selection methods and different numbers of cue-phrases for the addressee identification task. This is done on the filtered corpus and with the lexical filter.*

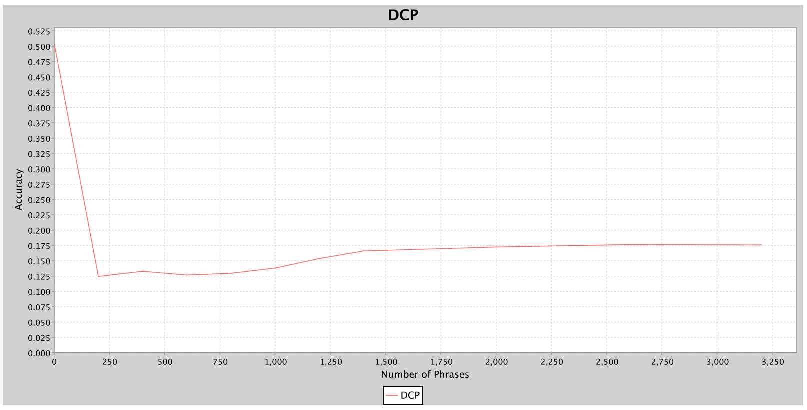


Figure 5.1: *Results of lexically filtered cue-phrase selection, for the addressee identification task, using the DCP method on the filtered addressee set, using NaiveBayes as learner.*

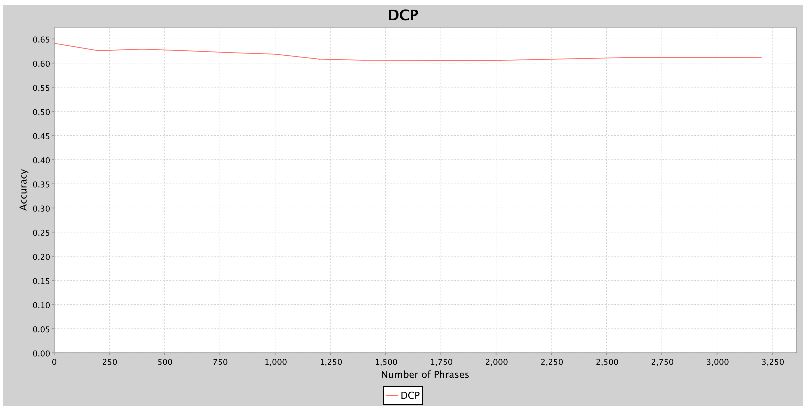


Figure 5.2: *Results of lexically filtered cue-phrase selection using the DCP method, for the addressee identification task. Using the filtered addressee set, with the J48 classifier as learner. With as extra features the Dialogue act for the utterance and the length of the utterance.*

COOC

	0	1	2
0	268	163	6
1	182	343	7
2	22	9	0

CP

	0	1	2
0	287	147	6
1	194	323	7
2	23	12	1

D

	0	1	2
0	299	148	6
1	202	301	12
2	19	10	3

DCP

	0	1	2
0	290	143	3
1	217	291	6
2	34	16	0

ENT

	0	1	2
0	15	449	0
1	8	484	0
2	0	44	0

IG

	0	1	2
0	276	141	5
1	202	322	12
2	22	19	1

MI

	0	1	2
0	274	139	3
1	206	331	7
2	21	19	0

S

	0	1	2
0	325	84	0
1	330	221	2
2	35	3	0

TTEST

	0	1	2
0	255	193	0
1	121	390	0
2	25	16	0

Table 5.6: *Confusion Matrixes for the addressee classification task. Shown for all cue-phrase selection methods used for the points where 3200 cue-phrases where selected. J48 was used for learning. In the confusion matrixes 0 stands for Single addressed, 1 for Group addressed and 2 for Unknown.*

	0	1
0	236	110
1	96	158

Table 5.7: *The confusion matrix for addressee selection using DCP for cue-phrase selection to select 3200 phrases on the addressee corpus without unkown sentences. In this matrix 0 stands for single addressed and 1 for group addressed.*

CHAPTER 6

Conclusion

6.1 Discussions

At first we briefly touched on the subject of discussions in meetings in chapter 2. Using the definition of discussions from [Rie06]: *"A dialogue,* **related to a single statement or proposition***, on which not all participants agree."*(Emphasis mine.) an annotation manual was made. Using this annotation manual (which can be found in appendix B) several annotators marked where they thought the discussions where in several meetings from the AMI corpus.

Even with these annotations we had a small amount of data available, so a qualitative analysis was done. We were forced to conclude that the definition as used in the annotation manual doesn't yield a good inter-annotator agreement. The main disagreement on the annotation of discussions seems to be the type of disagreement needed for something to be a discussion. Some annotated discussions, while correctly annotated, are about differences in beliefs about situations or the world in general. What we would like to have annotated are discussions concerning a difference in opinion. Further work should be done on a better annotation manual, taking into account that a more precise definition of the exact start and end point of a discussion is needed as well as the precise type of disagreement and dialogue necessary for it to qualify as a discussion. We propose a new definition of discussion to use for annotation: *"A dialogue, related to a single statement or proposition, on which not all participants agree. The disagreement should be a difference of opinion, and not a difference in beliefs about the world."*.

The results obtained from the discussions show no cue-words that occur significantly more frequently inside discussions than outside of them. However certain AMI Dialogue Acts seem to occur at the start and end of discussions. For the start of discussions these are suggest, asses, inform, Elicit-asses and Elicit-inform. For the end of discussions these are Stall, back-channel, Comment-About-Understanding, asses and sometimes inform. Other statistics that where looked at showed little significant changes, except the number of group-addressed sentences, which is higher outside discussions. Still group-addressed sentences make up only a small part of the corpus (around 30%) and the difference inside and outside discussions is not very large.

On the basis of this new definition of discussions, quite a lot of research is possible. First of all, it would need to be determined if this new definition, when coupled with an improved annotation manual, improves inter-annotator agreement. Should this be successful, one can start to try and teach a computer where discussions start and end.

6.2 Cue-phrases: Recipe and Addressee

In chapter 4 we introduced the recipe sentence classification task and in chapter 5 we introduced the single or group addressed addressee identification task. For both these task we used the cue-phrase selection methods as they where proposed in [SCVS99], we discussed them in detail in chapter 3. To be able to efficiently conduct multiple experiments using these cue-phrase selection methods we constructed to tool MAW and implemented the cue-phrase selection methods as feature generators within that tool. The final version of MAW, as it was written during the final project, performed very well on varied experiments producing the expected results. MAW also is easily customizable to use various corpora and features. We introduced the need for MAW in section 3.3 and briefly discussed its construction. For a more detailed manual, refer to appendix A.

In chapter 4 we presented the results we achieved on the recipe classification task. We have achieved an accuracy level of a little over 80%, which is comparable to the results as presented in [TA06] and [SCVS99]. And we have shown that several of the cue-phrase selection methods show the desired behavior. To reiterate the desired behavior is for the first n selected cue-phrases to contribute more to the accuracy achieved than cue-phrases which are selected after the first n. A clear example of this bahavior can be seen in figure 4.2. the methods showing the desired behavior are COOC, CP, D, DCP, MI and IG. The ENT, S and TTEST cue-phrase selection methods did not show the desired behavior.

In chapter 5 we presented the results that we achieved on single or group addressee identification task. The results that where achieved where not as succesfull. As can be seen in our best performance only achieved an accuracy of around 64%. On the other hand Jovanovic achieved very good results (83.7%) in her research ([NJN06]), however she used a number of computationally expensive features such as the gaze and gesture of the participants. Even when adding the dialogue act and the length of the utterance as features we did not achieve the desired performance. The COOC, CP, D , DCP, MI and IG cue-phrase selection methods did show the desired behavior.

The question that we posed: *"Can we determine if a phrase is single or group addressed based on automatically selected cue-phrases?"* can be answered with a clear no, as our best result only achieved an accuracy of around 64%. The baseline result lies around 38% or 50% (depending on the experiment), so there is an improvement of 26% or 14%. It seems that body language, including but not limited to the gaze of the speaker and the hand motions, plays an important part in the structuring of meetings c.q. conversations (as was previously remarked also in [NJN06, Gof81]). This statement is further supported by the great results achieved by Jovanovic when using the gaze and gesture information. Further research on this topic is needed as it would help quite a bit if, a selection could be made separating single and group addressed utterances, in a way that uses computationally cheaper features. Also, different cue-phrase selection methods could be tested, although the likelihood of an improvement is slim.

There are still possibilities left unexplored that might allow some aid in the single or group addressed classification task taht are, computationally, cheap. Features such as: the F0, speaker overlap, discussion state and Part-of-Speech tags or n-grams of part-of-speech tags, that could prove useful. The part-of-speech n-grams where used successfully both by Alofs and Latour in [TA06] as well as by Verbree in [Ver06].

The results achieved using the NaïveBayes classifier on both the recipe and the addressee task do not show the desired behavior. We conclude that the NaïveBayes classifier has problems dealing with cue-phrases as features, but we can offer no solid reason why this problem exists. However others, such as Yang, also noted the poor performance of the NaïveBayes classifier. In [Yan99] Yang noted a poor performance of NaïveBayes compared to other algorithms on a large dataset, though he could not offer an explanation either. For his experiments he used the Reuters dataset.

So, to summarize, the COOC, CP, D, DCP, MI and IG cue-phrase selection methods provide the behavior desired from cue-phrase selection methods. Cue-phrases selected with these methods work very well on the recipe sentence classification task. Depending on the use of lexical filtering CP,D,IG and MI work very well, D being the one that only works well without lexical filtering. Cue-phrases selected with these methods do not work well on the single or group addressed identification classification task, it needs more than just lexical features. Depending on the use of lexical filtering COOC, CP, D, DCP and MI work very well, DCP being the one that only works well when filtered. Like on the recipe task NaïveBayes classifiers produce extremely poor results when used with cue-phrases as features, we don't know why but similar behavior has been observed in [Yan99]. It is interesting to note that the D method shows good results, while in [SCVS99] it performs poorly. While it is not the best method overall, I would prefer the CP method due to its computational simplicity and overall decent performance. Overall, cue-phrases are very useful for suitable classification tasks and they should be applicable to a wide range of text classification problems.

With the research presented here, the base for further studies on cue-phrases is laid. First of all, the suitability of the different cue-phrase selection methods for different classification tasks can be examined. The measure of badness in the form of the D and DCP cue-phrase selection methods from [SCVS99] can be explored further, for example in the form of new cue-phrase selection methods. The combination of cue-phrase selection methods with different features as well as the notion of "compression" of the selected cue-phrases, as used by Verbree in [Ver06], can be explored using different cue-phrase selection methods. Another part that could be researched is the influence of different classifiers on the accuracy. In this thesis we mainly used the J48 classifier and we used the NaïveBayes classifier for a few experiments. It might also be usefull to check if the NaïveBayes classifier does perform better when using compression.

Bibliography

[AMI] AMI. Guidelines for dialogue act and addressee annotation version 1.0. AMI document on dialogue act and addressee annotation.

[Gof81] E. Goffman. *forums of talk*, chapter footing, pages 124–159. University of Pennsylvania Press, Philadelphia, 1981.

[IHW05] Eibe Frank Ian H. Witten. *Data Mining: Practical Machine Learning Tools and Techniques (Second Edition).* Morgan Kaufmann, San Francisco, 2005. ISBN: 0-12-088407-0, Weka can be found at `http://www.cs.waikato.ac.nz/ml/weka/`.

[JM00] Daniel Jurafsky and James H. Martin. *SPEECH and LANGUAGE PROCESSING An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* Prentice-Hall, 2000.

[Kat06] Hilco Kats. Classification Of User Utterances In Question Answering Dialogues. Master's thesis, University of Twente, 2006.

[MGS04] Julie Hirchberg Michel Galley, Kathleen McKeown and Elizabeth Shriberg. Identifying agreement and disagreement in conversational speech: Use of bayesian networks to model pragmatic dependencies. *Proc. of 42nd Meeting of the ACL*, 2004.

[NJN06] Rieks op den Akker Natasa Jovanovic and Anton Nijholt. Addressee identification in face-to-face meetings. unpublished, 2006.

[R$^+$] Peter Reutemann et al. Programmatic use. `http://weka.sourceforge.net/wiki/index.php/Programmatic_Use`.

[Rat97] A. Ratnaparkhi. A simple introduction to maximum entropy models for natural language processing. Technical report, nstitute for Research in Cognitive Science, University of Pennsylvania, 1997.

[Rie06] Rutger Rienks. Argumentation: its Role, its Rationality and its Structure. Internal document HMI group, 2006.

[RR06a] Daan Verbree Rutger Rienks. About the Usefulness and Learnability of Argument-Diagrams from Real Discussions, 2006.

[RR06b] Erik van der Weijden Rutger Rienks, Dirk Heylen. Argument Diagramming of Meeting Conversations. *Workshop at the 7th international conference on multimodal interfaces*, 2006.

[SCVS99] Ken Samuel, Sandra Carberry, and K. Vijay-Shanker. Automatically selecting useful phrases for dialogue act tagging, 1999.

[TA06] Jeroen Latour Thijs Alofs. Automatic classification of sentences in the recipe domain using a maxent classifier. unpublished, 2006.

[vdW05] Erik van der Weijden. Structuring argumentation in meetings. Master's thesis, University of Twente, 2005.

[vE04] Frank van Eynde. Part of speech tagging en lemmatisering van het corpus gesproken nederlands, 2004.

[Ver06] A. T. Verbree. On the structuring of discussion transcripts based on utterances automatically classified. Master's thesis, University of Twente, 2006.

[WXK05] Jonathan Kilgour Weiqun Xu, Jean Carletta and Vasilis Karaiskos. Coding Instructions for Topic Segmentation of the AMI Meeting Corpus, 2005. AMI document on topic segmentation.

[Yan99] Yiming Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1/2):69–90, 1999.

APPENDIX A

A Guide to MAW

MAW (Maw And Weka) is a relatively simple tool for the automatic extraction of features from corpora and training on the extracted features. For the training part MAW uses the WEKA machine learning toolkit [IHW05], this ensures a large amount of ML algorithms can be used easily. It also supports iterating over several arguments for the feature extractors and the trainingsize. Afterwards it is possible to generate a nice plot in postscript format, for easy inclusion in your documents.

In this appendix the user-side of things: running the tool, writing a suitable configuration file and adding your own corpusreaders and feature extractors will be discussed first. Afterwards the general program structure will be discussed to make it easier to modify or extend the program.

A.1 running maw

Maw depends on several external packages, a recent version of these packages should be available in the classpath. Maw uses: weka, org.jfree.chart, org.jibble.epsgraphics and org.apache.commons.math.

To run maw you need to compile the source-code and then run the MAW class (project.maw.MAW) with as single argument the location of the config file you wish to use. The format of this config file is described in the next section.

A.2 the config file

The configuration file is an plaintext (ASCII) file for which one may use either windows-style or unix-style line terminators. It contains four sections, each with a number of mandatory fields. The begin of a section is denoted by the name of that section on a line surrounded by square brackets. All other lines are in the option=value format. For all section all possible fields will now be described.

A.2.1 general

corpus Mandatory field, should contain the path to the corpus you wish to use. (how this is used depends entirely on the corpusreader implementation used)

corpustype Mandatory field, should contain the name of the class to use as corpusreader. This class should implement project.maw.readers.CorpusReader .

corpusargs Mandatory field, arguments to pass to the Corpus class. This is used at the moment only used as a way to generate an index of the corpusfeatures. This should be a colon separated list of the names of the corpusitem fields. This is useful for having features know what field of a corpusitem to use for feature generation. For example, the recipe-corpus used during testing contained 3 fields per line; The type of the utterance, the utterance itself and the POS-tags of the words in the utterance. In the testconfig the corpusargs value was "type:sent:pos". In this way the features could later know which field to use without having the order hardcoded in them.

outfile Mandatory field, points to a file where the information about the run should be saved. This overwrites any file already present.

plot Mandatory field, contains "yes" or "no", if yes the system will also create a nice plot, if no it won't.

plottoplot Mandatory if plot=yes, Denotes what to plot. Can use values used for features or for the trainingsize during the iterations as input. This is semi-colon separated per line plotted. For each line it is a comma-seperated list of items and a comma-seperated list of values to use while plotting.

plotrestvals Mandatory if plot=yes, semi-colon separated per line that is plotted. each line that is plotted contains a vertical-bar separated bar of var:value, which sets that variable to that value during plotting. This should set all the items iterated over that aren't used for plotting. Must have the same number of "lines" as plottoplot!

plottitle Mandatory if plot=yes, the title of the plot.

plotlinelabels Mandatory if plot=yes, a semi-colon separated list of line-labels. these line-labels will appear in the legend of the plot. Must have the same number of lines as plottoplot!

plotxlbl Mandatory if plot=yes, the label of the x-axis.

plotylbl Mandatory if plot=yes, the label of the y-axis.

plotfile Mandatory if plot=yes, the file where the plot should be saved (it will be saved as a postscript picture)

usesparseinstances "yes" or "no", default value is no. Denotes weather or not to use sparse-instances. See the weka documentation on weka.core.SparseInstance to see if this is a good idea. No is recommended unless you have a large number of features which will likely be 0 in value.

A.2.2 features

The features part of the config file is weird, it should contain 3 lines per feature used. (A feature is here a class that implements project.maw.features.Feature). The first part of each line (before the equals or underscore) is the name of the feature as it is referred to internally (for example in the plotting or iteration config file items).

$FNAME The name of the class that implements project.mnaw.features.Feature

$FNAME_args Arguments to give the feature, depends on the feature implementation, but for example the name of the corpusfeature to generate on is useful.

$FNAME_name The external name for the feature. Not particularly useful, make it whatever you want.

A.2.3 training

testsize Mandatory, the size of the testing set.

totrain Comma-separated list of features to be used for training, completely ignored at the moment (all features are always used). Quite easy to implement though.

tolearn Mandatory, what feature to use as the target for training, value must be an $FNAME.

learner Mandatory, the classifier to use, classifier must be a valid Weka classifier.

learnerargs A space-seperated list of arguments to pass to the learner.

A.2.4 iteration

Iteration is again special, it contains two special fields, and any number of generic iteration lines.

iterations Mandatory, the number of iterations to make over unchanging parameters. Must be 2 or larger, due to the nature of the statistics.

trainsizes Mandatory, a comma seperated list of trainingsizes to use.

A generic iteration item contains a comma separated list of $FNAME's , an equals sign and a comma separated list of values to set it to. Each iteration these values are passed (as string), to the relevant feature.

A.3 example configuration file

```
[general]
corpus=/Users/qadn/Desktop/Master_thesis/Recepten-Corpus.txt
corpustype=project.maw.readers.RecipeCorpusReader
corpusargs=type:sent:pos
outfile=/Users/qadn/Desktop/testout.txt
plot=yes
confidence=0.95
confusionmatrix=yes
```

```
plottoplot=trainingsize:10,100,1000;trainingsize:20,200,2000
plotrestvals=frop:frap;frop:frap
plottitle=plot
plotlinelabels=tientallen;twintigtallen
plotxlbl=xlbl
plotylbl=ylbl
plotfile=/Users/qadn/Desktop/plottest.eps
[features]
F0=project.maw.features.NumWordsFeature
F0_args=sent
F0_name=NumWords
F1=project.maw.features.RecipeToLearnFeature
F1_args=type
F1_name=RecipeType
[training]
testsize=1000
tolearn=F1
learner=weka.classifiers.trees.J48
learnerargs=
[iteration]
iterations=5
trainsizes=10,20,100,200,1000,2000
```

A.4 Creating your own features and corpusreaders

To create features (classes implementing project.maw.features.Feature) and corpusreaders, simply implement these interfaces and use them in your config file. The interface implementation is discussed here.

A.4.1 project.maw.readers.CorpusReader

Vector<CorpusItem> readin(String corpus, String[] features)

This is the only method a corpusreader is required to implement. The corpus string points at the location of the corpus (as particular to the corpusreader implementation) and the String[] features contains the names of all the features one wants to read in from the corpus (you might not want them all due to space considerations). The corpusreader then creates a Vector containing CorpusItems, each corpusitem represents one unit from the corpus. Each corpusitem should contain the exact number of features given in the String[] in the proper order.

A.4.2 project.maw.features.Feature

void init(Corpus corpus, String args, String name, String fname)

Should initialize the feature. corpus refers to the corpus. args refers to the args as given in the config file. name refers to the name as given in the config file. fname refers to the name as used in the config file (the $FNAME). Any and all initialization needed should be done here, the implementation of the args argument is Feature specific and should be specified in the javadoc.

void iter(String arg)

This method is called each time a new iteration starts that also iterates on this feature, the arg contains the iteration arguments. The handling of the arg argument is Feature specific and should be explained in the javadoc. This method should prepare the Feature for a new iteration.

Object[] gen(CorpusItem item)

Generate all values given a certain corpusitem. For all weka features this Feature produces the returned array should contain one object. These object should be of the type String if this is a nominal Feature, or of the type Double if it is a numeric Feature.

boolean isNumeric()

Denotes if this Feature generates numeric weka features.

boolean isNominal()

Denotes if this feature generates nominal weka features.

Vector<String> getPossibleValues()

Returns all possible values this nominal feature can take, only called if it generates a single weka feature AND if it is nominal. If the Feature isn't nominal do as you please here (return null or whatever)

Vector<String> getPossibleValues(int index)

Returns all possible values of the nominal feature at the index. The index starts at 0. If this Feature isn't nominal do as you please.

String getName()

Return the name of the feature.

String getName(int index)

Return the name of the feature at index. Indexes start at 0.

String getFName()

Returns the fname of the feature. The name the feature is referred to in the config file.

boolean isMulti()

If your feature generates multiple weka features return true here.

int numFeatures()

Return the number of features this feature implements. This value may only change once iter(String args) is called. The Feature must always generate exactly the number of weka features as this function says it does.

A.4.3 Example feature implementations

In this section the implementation of two very simple Features, one generates two numeric weka features, one generates a single nominal weka feature.

```
package project.maw.features;

import java.util.Vector;

import project.maw.corpus.Corpus;
import project.maw.corpus.CorpusItem;

/**
 * A simple feature denoting the number of words in a sentence.
 * The FNAME_args=bla should contain the name of the corpusfeature to train on!
 * (corpus.getIndexOfFeature(args) should refer to the position of the relevant part in a corpusitem.
 * So corpusitem.getFeature(corpus.getIndexOfFeature(args));
 * Should return the sentence to test the number of words on.
 * Implements an additional feature: the avarage length of the word. (used to test this functionality)
 */

public class NumWordsFeature implements Feature{

  private Corpus corpus;
  private String name;
  private String fname;
  private String args;
  private int index;

  public Object[] gen(CorpusItem item) {
    // not applicable: null
    Object[] out = new Object[2];
    out[0] = new Double(""+item.getFeature(index).split(" ").length);
    String[] frop = item.getFeature(index).split(" ");
    int total = 0;
    for(int i = 0; i < frop.length; i++)
    {
      total += frop[i].length();
    }
    out[1] = new Double(((double) total) / ((double) frop.length));
    return out;
  }

  public String getFName() {
    return fname;
  }

  public String getName() {
    return null;
  }

  public String getName(int index)
  {
    return name+index;
  }

  public Vector<String> getPossibleValues() {
    return null;
  }

  public void init(Corpus corpus, String args, String name, String fname) {
    this.corpus = corpus; // the corpus
    this.name = name; // the name of the feature
    this.fname = fname; // the name the feature is referred to in the corpus
    this.args = args; // the arguments in the config file.
    index = corpus.getindexoffeature(args);
  }

  public boolean isNominal() {
    return false;
  }

  public boolean isNumeric() {
    return true;
  }

  public void iter(String arg) {
    return;
  }

  public Vector<String> getPossibleValues(int index) {
    return null;
  }

  public boolean isMulti() {
    return true;
  }

  public int numFeatures() {
    return 2;
  }
}
```

```
package project.maw.features;
```

```
import java.util.Vector;

import project.maw.corpus.Corpus;
import project.maw.corpus.CorpusItem;

/**
 * An extremely simple feature to be able to lean the recipe things.
 * simple gives the to learn thing.
 */

public class RecipeToLearnFeature implements Feature {

  private Corpus corpus;
  private String name;
  private String fname;
  private String args;
  private int index;

  public Object[] gen(CorpusItem item) {
    Object[] out = new Object[1];
    out[0] = item.getFeature(index);
    return out;
  }

  public String getFName() {
    return fname;
  }

  public String getName() {
    return name;
  }

  public String getName(int index)
  {
    return null;
  }

  public Vector<String> getPossibleValues() {
    Vector<String> out = new Vector<String>();
    out.add("requirement");
    out.add("signal");
    out.add("ins-block");
    out.add("emphasis");
    out.add("ins-nonblock");
    out.add("suggestion");
    out.add("explanation");
    out.add("inf-irrelevant");
    out.add("inf-relevant");
    out.add("name");
    out.add("other");
    return out;
  }

  public void init(Corpus corpus, String args, String name, String fname) {
    this.corpus = corpus; // the corpus
    this.name = name; // the name of the feature
    this.fname = fname; // the name the feature is referred to in the corpus
    this.args = args; // the arguments in the config file.
    index = corpus.getindexoffeature(args);
  }

  public boolean isNominal() {
    return true;
  }

  public boolean isNumeric() {
    return false;
  }

  public void iter(String arg) {
    return;
  }

  public Vector<String> getPossibleValues(int index) {
    return null;
  }

  public boolean isMulti() {
    return false;
  }

  public int numFeatures() {
    return 1;
  }
}
```

A.5 Program Structure

To illustrate the program structure a simple run of maw will be discussed in detail, this should provide you with an insight to which class does what. Also a picture containing the most important classes for a visual overview is given in

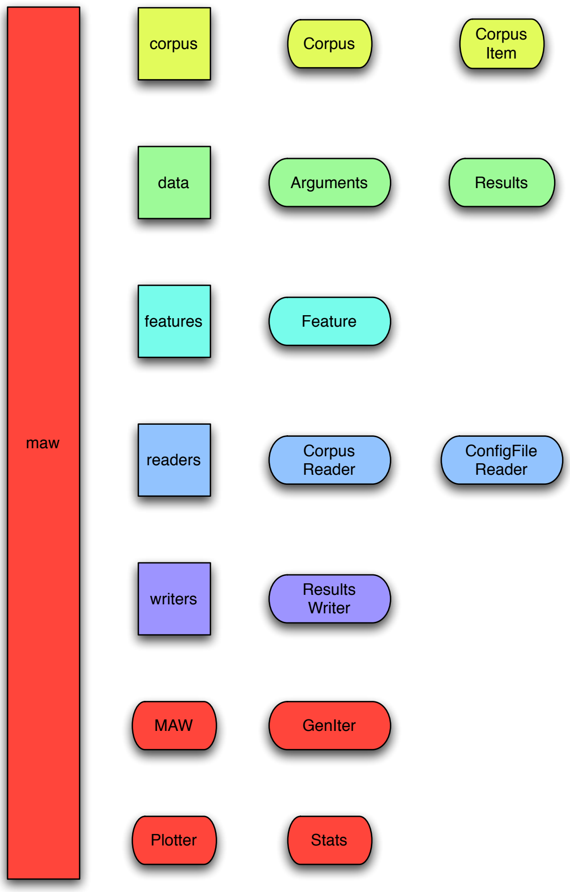


Figure A.1: *The structure of the maw program, only lists the most important classes in each package. Packages are denoted with square boxes, classes with rounded boxes.*

figure A.1.

To begin the main method in the class MAW initializes a MAW class instance with a String pointing to a config file. The Maw constructor then creates a new readers.ConfigFileReader, that reader then reads in the config file and returns the configuration as a data.Arguments instance.

Then the corpus is read in by creating a corpus.Corpus class and asking it to read in the corpus. The Corpus class initializes the correct (as denoted in the config) readers.CorpusReader and stores a Vector of corpus.CorpusItem.

Then all features.Feature extending classes (as denoted in the configuration) are instantiated with the arguments from the configuration file. The GenIter class is instantiated, this class generates all iterations on the basis of the config file, also it works as a sort of java.util.Iterator over the iterations. Various usefull classes are instantiated then; Stats, a small class that is used for computing very basic statistics (basicly the confidence interval and mean of a series of numbers); Results, a class for storing results in; ResultsWriter, a class that writes away the results to a file; And finally Plotter, a class that is used for plotting the results is instantiated.

When all these classes are instantiated the main loop is started. This loop does the following for each defined iteration; Retrieve the configuration for the current iteration from the GenIter instance; Pass all initialized feature.Feature classes their iteration specific arguments; Generate a Weka-friendly version of the training and testing set using weka.core.Instance, see also [R$^+$]; Train with the weka-compatible classifier as given in the config file; Test on the testset.

If applicable the results are plotted via the MAW.plot() method, and finally all results are stored via the MAW.store() method.

APPENDIX B

Annotation Guide

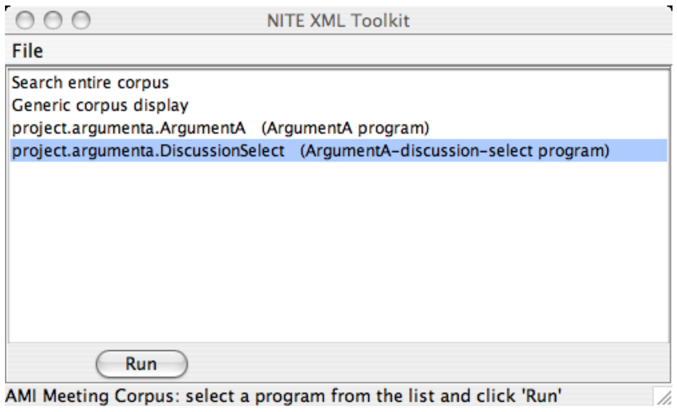
B.1 Introduction

Lately there has been some research into the visualisation and annotation of discussions in meetings at the University of Twente. Several people at the HMI dept. of the University of Twente have been involved in the development of TAS, a discussion annotation scheme with a visualisation thereof. However something that was not adressed clearly, and isn't adressed clearly in the literature either is when discussions start and stop. To answer this question a rough definition is given and we will try to see if a number of annotators agree with eachother on this.

B.2 The Task: Discussion Segmentation

We shall define discussion as: *A dialogue,* **related to a single statement or proposition**, *on which not all participants agree.*

You as an annotator, should segment the discussions to the best of your ability. There is not a minimum or maximum number of discussions in a meeting, although there is a practical limit offcourse. Neither is the lenght of a discussion constrained, it can be as short as two lines or as long as the meeting. Not every utterance in the meeting has to be part of a discussion. The granularity of the discussion selection is that of an utterance. Parts of discussions can overlap, for example the last two utterances of one discussion might be the first two utterances of the next discussion. The meetings IS1003a, IS1003b, IS1003c and IS1003d should be annotated. Depending on ones reading and annotation speed this could take anywhere from half an hour to an hour per meeting. (inspired by [WXK05])



Search entire corpus
Generic corpus display
project.argumenta.ArgumentA (ArgumentA program)
project.argumenta.DiscussionSelect (ArgumentA-discussion-select program)

AMI Meeting Corpus: select a program from the list and click 'Run'

Figure B.1: *Selecting which Argumenta program to run.*

B.3 A Brief Annotation Tool Guide.

B.3.1 Installing and Running the Tool

This assumes you have a working java 1.5 installation on you computer. The "Argumenta Standalone.zip" is available from kletsmajoor in the /local/data/usr/stehouwe directory. Extract the zip file. On unix-like platforms run start.sh from the directory you extracted the zip file to. On windows run the start.bat from the directory you extracted the zip file to.

B.3.2 Using the Tool

In this section it will be described how to use the annotation tool. This assumed an already setup and working Argumenta installation.

1. First of all start Argumenta.
2. You will be presented with a dialog screen asking you to select which Argumenta program to run. See fig. B.1. Here choose project.argumenta.DiscussionSelect.
3. Next you will see an annotator selection screen, like fig. B.2. Choose your name or make a new annotator in your name.
4. After selecting the annotator you will be asked to select the meeting you whish to annotate, much like in figure B.3. Here select the meeting you whish to annotate.
5. Now the discussion select tool starts. When it is done loading you should see a screen roughly like figure B.4. You might have to move some windows around to get the same effect.
6. You should now read the transcription in the transcription window carefully before continueing.
7. In this transcription window you should now select each discussion you encountered during your read and do the following steps.

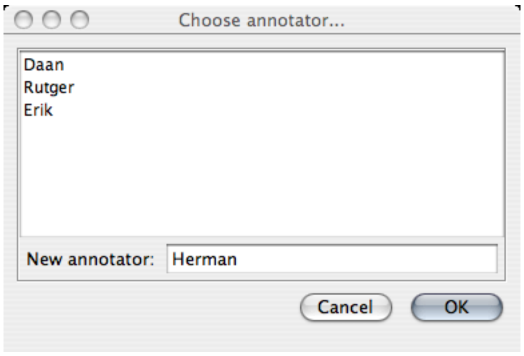


Figure B.2: *Selecting which annotator is annotating, or alternativly defining a new annotator.*

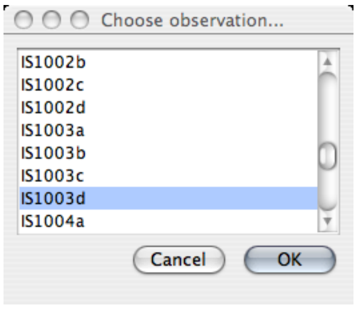
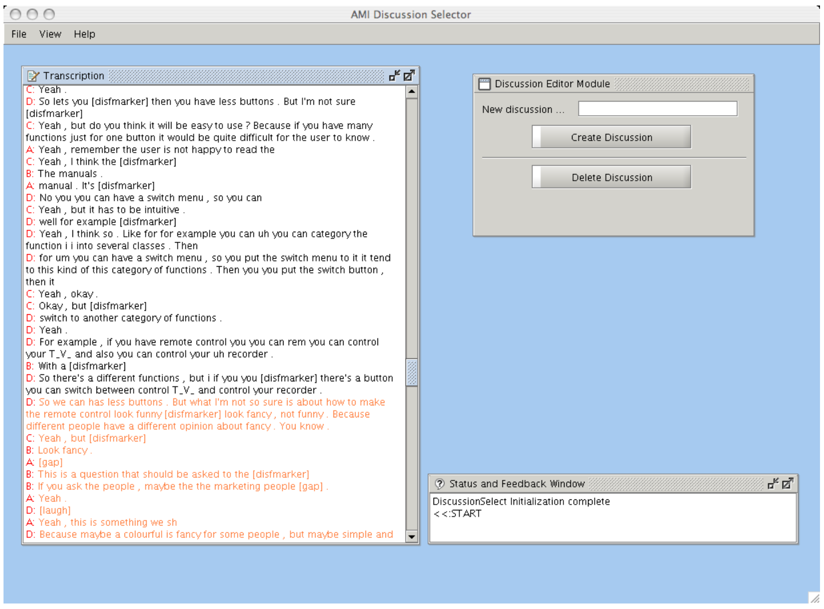


Figure B.3: *Selecting which meeting to annotate.*



C: Yeah .
D: So lets you [disfmarker] then you have less buttons . But I'm not sure [disfmarker]
C: Yeah , but do you think it will be easy to use ? Because if you have many functions just for one button it would be quite difficult for the user to know .
A: Yeah , remember the user is not happy to read the
C: Yeah , I think the [disfmarker]
B: The manuals .
A: manual . It's [disfmarker]
D: No you you can have a switch menu , so you can
C: Yeah , but it has to be intuitive .
D: well for example [disfmarker]
D: Yeah , I think so . Like for for example you can uh you can category the function i i into several classes . Then
D: for um you can have a switch menu , so you put the switch menu to it it tend to this kind of this category of functions . Then you you put the switch button , then it
C: Yeah , okay .
C: Okay , but [disfmarker]
D: switch to another category of functions .
D: Yeah .
D: For example , if you have remote control you you can rem you can control your T_V_ and also you can control your uh recorder .
B: With a [disfmarker]
D: So there's a different functions , but i if you you [disfmarker] there's a button you can switch between control T_V_ and control your recorder .
D: So we can has less buttons . But what I'm not so sure is about how to make the remote control look funny [disfmarker] look fancy , not funny . Because different people have a different opinion about fancy . You know .
C: Yeah , but [disfmarker]
B: Look fancy .
A: [gap]
B: This is a question that should be asked to the [disfmarker]
B: If you ask the people , maybe the the marketing people [gap] .
A: Yeah .
D: [laugh]
A: Yeah , this is something we sh
D: Because maybe a colourful is fancy for some people , but maybe simple and

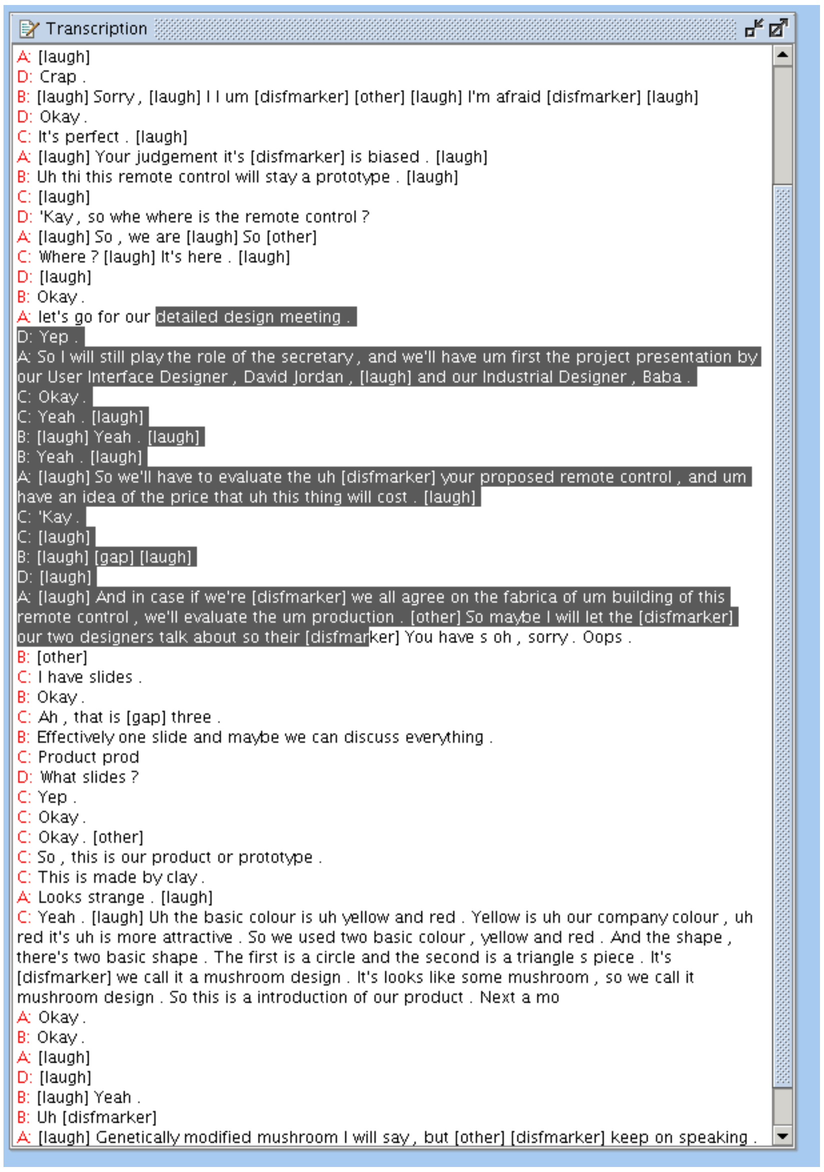
DiscussionSelect Initialization complete
<<:START

Figure B.4: *An overview screenshot of the argumenta discussion select tool; notice that already marked discussions are orange.*

   (a) Select the discussion in the transcription window like in figure B.5.
   (b) Fill in a discussion name in the discussion create window, like in figure B.6.
   (c) Click on “Create Discussion”.

8. Click on “File” and then on “Save”.
9. Exit the tool.

After this is done for all meetings you wish to Annotate please pass me the contents of the NXT-format/xml directory.



A: [laugh]
D: Crap .
B: [laugh] Sorry , [laugh] I I um [disfmarker] [other] [laugh] I'm afraid [disfmarker] [laugh]
D: Okay .
C: It's perfect . [laugh]
A: [laugh] Your judgement it's [disfmarker] is biased . [laugh]
B: Uh thi this remote control will stay a prototype . [laugh]
C: [laugh]
D: 'Kay , so whe where is the remote control ?
A: [laugh] So , we are [laugh] So [other]
C: Where ? [laugh] It's here . [laugh]
D: [laugh]
B: Okay .
A: let's go for our detailed design meeting .
D: Yep .
A: So I will still play the role of the secretary , and we'll have um first the project presentation by our User Interface Designer , David Jordan , [laugh] and our Industrial Designer , Baba .
C: Okay .
C: Yeah . [laugh]
B: [laugh] Yeah . [laugh]
B: Yeah . [laugh]
A: [laugh] So we'll have to evaluate the uh [disfmarker] your proposed remote control , and um have an idea of the price that uh this thing will cost . [laugh]
C: 'Kay .
C: [laugh]
B: [laugh] [gap] [laugh]
D: [laugh]
A: [laugh] And in case if we're [disfmarker] we all agree on the fabrica of um building of this remote control , we'll evaluate the um production . [other] So maybe I will let the [disfmarker] our two designers talk about so their [disfmarker] You have s oh , sorry . Oops .
B: [other]
C: I have slides .
B: Okay .
C: Ah , that is [gap] three .
B: Effectively one slide and maybe we can discuss everything .
C: Product prod
D: What slides ?
C: Yep .
C: Okay .
C: Okay . [other]
C: So , this is our product or prototype .
C: This is made by clay .
A: Looks strange . [laugh]
C: Yeah . [laugh] Uh the basic colour is uh yellow and red . Yellow is uh our company colour , uh red it's uh is more attractive . So we used two basic colour , yellow and red . And the shape , there's two basic shape . The first is a circle and the second is a triangle s piece . It's [disfmarker] we call it a mushroom design . It's looks like some mushroom , so we call it mushroom design . So this is a introduction of our product . Next a mo
A: Okay .
B: Okay .
A: [laugh]
D: [laugh]
B: [laugh] Yeah .
B: Uh [disfmarker]
A: [laugh] Genetically modified mushroom I will say , but [other] [disfmarker] keep on speaking .

Figure B.5: *Selecting a discussion, note that Argumenta will automaticly select in utterance units.*

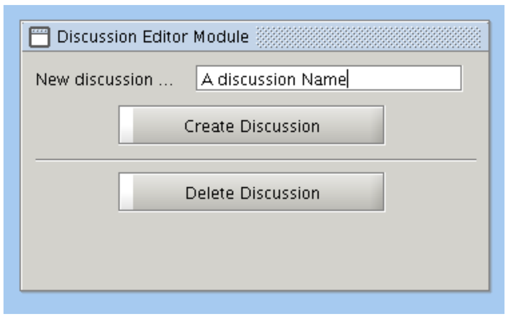


Figure B.6: *Creating a discussion.*

APPENDIX C

Start and End of Discussions

For each discussion this appendix contains a table containing the title of the discussion, as marked by the annotator, the Dialogue act of all start and ending sentences, as well as the addressee speaker and raw utterance corresponding to those dialogue acts. For this the dialogue act annotation of the annotator "vkaraisk" where used.

Please note that Addressee annotations where only available for the meetings IS1003b and IS1003d, out of the four discussions for which we have discussion annotation data as annotated with our annotation guide. So for meetings a and c the addressee is always denoted as Unknown by our script.

C.1 Start and End of the IS1003a Discussions

Begin Discussion, Annotator H				
Title	**DA**	**addr**	**Spkr**	**Sentence**
who's who	Stall	U	B	Okay . So

End Discussion, Annotator H				
Title	**DA**	**addr**	**Spkr**	**Sentence**
who's who	Inform	U	C	I'm the industrial designer .

Begin Discussion, Annotator J				
Title	**DA**	**addr**	**Spkr**	**Sentence**
1	Inform	U	A	The tool training is to try out the white board ,
2	Stall	U	A	So
3	Stall	U	A	so
4	Stall	U	D	So
5	Inform	U	C	I I think the user the user interface design is [disfmarker] he will design how the user will you know [disfmarker] the relation between the user and you know the remote control so [disfmarker]
6	Inform	U	A	As we want to maximise the benefit .
7	Stall	U	A	So ,
8	Elicit-Inform	U	B	[laugh] What's this ?
9	Stall	U	C	Yes
10	Elicit-Inform	U	B	A person ? [laugh]

End Discussion, Annotator J				
Title	**DA**	**addr**	**Spkr**	**Sentence**
1	Stall	U	C	Yeah ,
2	Assess	U	D	It's reasonable , I think , yeah .
3	Assess	U	A	Yeah .
4	Backchannel	U	D	Mm .
5	Backchannel	U	B	Okay .
6	Backchannel	U	B	Okay .
7	C-A-U	U	B	Yeah .

8	Assess	U	C	Yeah , it is a [disfmarker] [laugh]
9	Assess	U	B	[laugh] Yeah , it's okay .
10	Inform	U	B	[laugh] Dog . [laugh]

Begin Discussion, Annotator R

Title	DA	addr	Spkr	Sentence
diff. between UI and ID	Stall	U	B	Okay . So

End Discussion, Annotator R

Title	DA	addr	Spkr	Sentence
diff. between UI and ID	Stall	U	C	Okay . So

C.2 Start and End of the IS1003b Discussions

Begin Discussion, Annotator H

Title	DA	addr	Spkr	Sentence
whire-or-whireless	Stall	U	C	Yeah
less buttons	Assess	D	C	Yeah ,
interfacing	Offer	A,C,B	D	which I want to have sophisticated functions while with very easy to use user interface .

End Discussion, Annotator H

Title	DA	addr	Spkr	Sentence
whire-or-whireless	Backchannel	U	B	Okay [laugh] .
less buttons	Inform	A,C,B	D	So there's a different functions ,
interfacing	Assess	A	D	Yeah , that's the point .

Begin Discussion, Annotator J

Title	DA	addr	Spkr	Sentence
q	Elicit-Inform	D	C	Um , so you're participant two ?
w	Stall	U	C	'Kay ,
e	Stall	U	C	Yeah um
r	Elicit-Inform	B	C	So you mean that infrared control is a cheap technology ?
t	Stall	U	B	So
y	Inform	D,C,B	A	I don't think [disfmarker] well , yeah , I don't think he would ,
u	Inform	D,C,B	A	Um actually seventy five percent of the users find m the most remote controls uh ugly ,
i	Inform	D	B	The manuals .
o	Suggest	A,C,B	D	Maybe we can have di di we can have uh several options ,
p	Fragment	U	C	It's is uh [disfmarker]
a	Assess	C	B	[laugh] Yeah [laugh] . Yeah ,
s	Inform	B	D	It's It's not so uh popular now . [laugh]
d	Suggest	A,D,C	B	I think if you have you know th like a yellow ribbon here is the double R_ . Or should be .

End Discussion, Annotator J

Title	DA	addr	Spkr	Sentence
q	Backchannel	U	A	Okay .
w	Stall	U	D	Okay , so .
e	C-A-U	D	A	Okay .
r	Stall	U	B	Yeah ,
t	Fragment	U	D	[laugh]
y	Inform	D	C	I think it's more your problem .
u	Stall	U	A	So
i	Backchannel	U	B	Yeah , yeah , yeah .
o	Assess	C	B	Yeah .
p	Backchannel	U	A	Yeah .
a	C-A-U	D	B	Okay .
s	Backchannel	U	D	Okay .
d	Stall	U	C	Okay . So [disfmarker]

Begin Discussion, Annotator R

Title	DA	addr	Spkr	Sentence
powerful easy to use	Stall	U	A	So ,
wire remote control ?	Elicit-Inform	B	D	So what , the wireless remote control ?
how many buttons	Elicit-Assessment	D	C	And just to have uh an idea , do you think you as the User Interface Designer to would it be possible to have less buttons and still have the same functionality and to have powerful remote control , you think it's possible ?
fancy	Inform	A,C,B	D	So we can has less buttons .
the colour	E-O-o-S	C	D	As as for the colour , what what do you think ?

End Discussion, Annotator R

Title	DA	addr	Spkr	Sentence
powerful easy to use	Assess	A	D	Yeah , that's the point .
wire remote control ?	Stall	U	A	Well
how many buttons	Inform	A,C,B	D	So there's a different functions ,
fancy	Assess	U	A	Yeah ,
the colour	E-C-U	A,D,B	C	It's okay ?

C.3 Start and End of the IS1003c Discussions

Begin Discussion, Annotator H

Title	DA	addr	Spkr	Sentence
speech gesture	Offer	U	A	I just have one question ,
manual controller?	Stall	U	A	Yeah ,
cost	Stall	U	A	Yeah ,
smart controller	Stall	U	B	Well
voice box	Elicit-Inform	U	A	And what about voice recognition , do we have microphones ?
type of remote	Stall	U	A	Hmm . So ,
spongy	Inform	U	D	we go to the fruits and vede vegetables .
material	Elicit-Assessment	U	A	Something spongy . [laugh]

voice gest	Assess	U	A	Because I think that with the voice and gesture recognition there are still some disadvantages with this .
slides where	Suggest	U	B	Can you go to the [laugh] next one ?

End Discussion, Annotator H

Title	DA	addr	Spkr	Sentence
speech gesture	Backchannel	U	C	Yeah .
manual controller?	Suggest	U	B	manual controllers , eh .
cost	Inform	U	D	I mean this doesn't have uh the power to do recognition , for example .
smart controller	Assess	U	C	[laugh] Yeah ,
voice box	Assess	U	B	[laugh] Yeah , we should . [laugh] Uh .
type of remote	Backchannel	U	C	Yeah .
spongy	Stall	U	D	Okay ,
material	Assess	U	B	Okay .
voice gest	Assess	U	D	I dunno .
slides where	Inform	U	B	I can say it to you without .

Begin Discussion, Annotator J

Title	DA	addr	Spkr	Sentence
11	Assess	U	D	No
12	Stall	U	C	Yeah ,
13	Suggest	U	B	And mayb maybe we can have the switching mode to pass from you know voice controller to
14	Inform	U	A	The recognition system will be able to understand French .
15	Elicit-Inform	U	A	And with no increase in the pri production price of the remote control ?
16	C-A-U	U	A	Oh .
17	Suggest	U	D	But you would still have the buttons .
18	Elicit-Assessment	U	A	You think it's possible ?
19	Inform	U	A	You don't need to tune it .
20	Stall	U	D	I mean ,
21	Assess	U	A	Uh I think there's something wrong with your [disfmarker]
22	Stall	U	A	Yeah ,
23	Elicit-Assessment	U	D	What about the touch scr touch screen ? For example .
24	Inform	U	D	we go to the fruits and vede vegetables .
25	Stall	U	A	So
26	E-O-O-S	U	A	Case .
27	Backchannel	U	C	Yeah .
28	E-O-O-S	U	D	What interface ?

End Discussion, Annotator J

Title	DA	addr	Spkr	Sentence
11	Assess	U	C	Yep .
12	Assess	U	C	It's advantage .
13	Backchannel	U	D	Yeah .
14	C-A-U	U	A	Mm , okay .
15	Backchannel	U	C	Yeah .
16	Backchannel	U	B	Yeah .
17	Assess	U	B	[laugh] Yeah , we should . [laugh] Uh .
18	Backchannel	U	D	Yeah .
19	Backchannel	U	C	Yeah .
20	Backchannel	U	C	Yeah .
21	Assess	U	D	Yeah , true .
22	Assess	U	A	Okay .
23	Backchannel	U	D	Mm .
24	Backchannel	U	B	Okay .
25	Backchannel	U	B	Yeah .
26	Backchannel	U	D	Yeah .
27	Backchannel	U	C	Mm .
28	Assess	U	A	Okay .

Begin Discussion, Annotator R

Title	DA	addr	Spkr	Sentence
voice and gestures ?	Offer	U	A	I just have one question ,
what about materials ?	E-O-O-S	U	A	And what tha what about the uh materials ?
what about the buttons ?	Stall	U	A	So
voice recog in tv	Stall	U	B	Mm . Yeah .
is this a discussion?	Assess	U	A	So maybe we'll just focus on the Google controller plus the fancy controller ,

End Discussion, Annotator R

Title	DA	addr	Spkr	Sentence
voice and gestures ?	Backchannel	U	C	Yeah .
what about materials ?	Assess	U	A	Okay .
what about the buttons ?	Backchannel	U	D	Mm .
voice recog in tv	Assess	U	D	I dunno .
is this a discussion?	Assess	U	A	Okay .

C.4 Start and End of the IS1003d Discussions

Begin Discussion, Annotator H

Title	DA	addr	Spkr	Sentence
eval: fancy	Suggest	A,C,B	D	we gonna try to measure how good it is instead of just talking about [disfmarker]
eval: easy to use	Elicit-Assessment	A,C,B	D	[other] L last one w I would like to judge is is it easy to use ?
eval: kosten	Stall	U	A	[other] So
eval: tasks	C-A-U	A,D,C	B	So what are we going to do with this project evaluation ?
Pine Apple	Stall	U	A	Yeah
mushroom	Inform	D	A	Fruits and vegetables .
shapish	Stall	U	D	No ,

End Discussion, Annotator H

Title	DA	addr	Spkr	Sentence
eval: fancy	Other	U	D	Three .

eval: easy to use	Assess	D	B	Yeah ,
eval: kosten	Assess	C	B	It's fine , twelve fifty
eval: tasks	Backchannel	U	B	Okay , okay .
Pine Apple	Backchannel	U	C	Okay .
mushroom	Backchannel	U	C	Yeah . [laugh]
shapish	Assess	D	B	It's perfect ,

Begin Discussion, Annotator J

Title	DA	addr	Spkr	Sentence
30	Stall	U	C	So ,
31	Inform	A,D,B	C	the second key feature is that uh s circle channel um selection .
32	Stall	U	C	Yeah ,
33	Stall	U	C	Okay .
34	Stall	U	D	So
35	Elicit-Assessment	A,C,B	D	Do we have a fancy look and feel , according to you ?
36	Elicit-Assessment	D	B	The colour , is the colour acceptable ?
37	Elicit-Assessment	A,C,B	D	The other criterion is is it technologically uh technologically in innovative .
38	Elicit-Assessment	A,C,B	D	[other] L last one w I would like to judge is is it easy to use ?
39	Stall	U	A	yeah ,
40	Assess	D	C	Yeah
41	Elicit-Assessment	D	A	And what's your opinion ?
42	Assess	D	B	Yeah ,
43	Stall	U	A	[other] So
44	Stall	U	A	Uh
45	Elicit-Inform	D,C,B	A	Has it changed .
46	Elicit-Inform	A,D,B	C	Which part is the most expensive part ? [laugh]
47	Stall	U	A	Yeah
48	C-A-U	A,D,C	B	So what are we going to do with this project evaluation ?
49	Assess	A,D,C	B	But I think it's good to follow the f flow
50	Inform	D	C	Twenty five Euros ,
51	Assess	B	D	True .
52	Stall	U	C	So

End Discussion, Annotator J

Title	DA	addr	Spkr	Sentence
30	Backchannel	U	A	Okay .
31	Assess	A	C	Mushroom design , yeah .
32	Backchannel	U	C	Yeah .
33	Backchannel	U	C	Okay .
34	Backchannel	U	C	Okay .
35	Backchannel	U	D	Okay .
36	Other	U	D	Three .
37	Assess	D	B	Two , yeah , two .
38	Backchannel	U	C	Yeah . [laugh]
39	C-A-U	D,C,B	A	Okay .
40	Fragment	U	C	[laugh]
41	Stall	U	D	Yeah .
42	Assess	A	C	Yeah .
43	Backchannel	U	D	Yeah . [laugh]
44	Stall	U	D	Um ,
45	Stall	U	A	So , yeah .
46	Inform	A,D,B	C	Cheaper .
47	Backchannel	U	D	Yeah .
48	Backchannel	U	C	Yeah .
49	Backchannel	U	C	Yeah .
50	Backchannel	U	C	Okay .
51	Backchannel	U	C	Okay .
52	Backchannel	U	A	Yeah .

Begin Discussion, Annotator R

Title	DA	addr	Spkr	Sentence
mushroom design vegetable	Fragment	U	B	I don't think [disfmarker] [laugh]
what about look and feel?	Elicit-Assessment	A,C,B	D	Do we have a fancy look and feel , according to you ?
from two to channel twenty	Elicit-Assessment	A,C,B	D	[other] L last one w I would like to judge is is it easy to use ?
project accepted	Stall	U	A	So ,
a cute name for it	Suggest	A,D,B	C	So we have give him [disfmarker] give it a cute name .

End Discussion, Annotator R

Title	DA	addr	Spkr	Sentence
mushroom design vegetable	Inform	A,C,B	D	But anyway this is not a mushroom anyway ,
what about look and feel?	Inform	D,C,B	A	It looks [other] more like a pineapple .
from two to channel twenty	Stall	U	D	Okay
project accepted	Stall	U	A	[other] So uh mm mm . Mm .
a cute name for it	Backchannel	U	C	Okay .

APPENDIX D

Adjacency Pairs Inside and Outside Discussions

The distribution of Adjacency Pairs by speaker patterns as they are annotated on the AMI corpus. The first letter of the speaker pattern denotes the speaker of the first sentence (always A), the second letter denotes whom he is addressing the sentence to, this can either be "B" (to an individual) or X (to a group. The third and fourth letter denote the second sentence of the Adjacency Pair, again first the speaker and then the addressee. The speaker can be A (the same speaker), B (a new speaker or the participant addressed earlyer by A) or C a different speaker. The addressee works similar to the that of the first sentence (A if addressed to the speaker of the first sentence, B if addressed to the single addressee of the first sentence, C if addressed to a new person, D if addressed to a new person or X if addressed to a group). Horizontally the type of relation is denoted.

Please note that Addressee annotations where only available for the meetings IS1003b and IS1003d, out of the four discussions for which we have discussion annotation data as annotated with our annotation guide. So for meetings b and d more statistics are provided.

IS1003a

Not Available.

IS1003b

H

Inside Discussions

	NEG	PART	POS	UNC
ABBA	1	1	20	
ABBX				
ABBC				
AXBX			1	
AXBA	1		8	
AXBC			1	
ABCA			4	
ABCB				
ABCD			1	
ABCX			1	
AXAX				
ABAX			2	
ABAB				
AXAB				
ABAC				

Outside Discussions

	NEG	PART	POS	UNC
ABBA	2		42	1
ABBX			1	
ABBC				
AXBX			1	
AXBA	2		19	
AXBC				
ABCA	1		9	1
ABCB				
ABCD			3	
ABCX			2	1
AXAX		1	14	
ABAX			1	
ABAB				
AXAB				
ABAC				

J

Inside Discussions

	NEG	PART	POS	UNC
ABBA	2	1	30	1
ABBX			1	
ABBC				
AXBX			2	
AXBA	2		14	
AXBC			1	
ABCA	1		7	1
ABCB				
ABCD			3	
ABCX			2	1
AXAX		1	6	
ABAX			1	
ABAB				
AXAB				
ABAC				

Outside Discussions

	NEG	PART	POS	UNC
ABBA	1		32	
ABBX				
ABBC				
AXBX				
AXBA	1		13	
AXBC				
ABCA			6	
ABCB				
ABCD			1	
ABCX			1	
AXAX			8	
ABAX			2	
ABAB				
AXAB				
ABAC				

R

Inside Discussions

	NEG	PART	POS	UNC
ABBA	1	1	21	1
ABBX			1	
ABBC				
AXBX			1	
AXBA	3		12	
AXBC			1	
ABCA			5	
ABCB				
ABCD			3	
ABCX			2	1
AXAX				
ABAX			2	
ABAB				
AXAB				
ABAC				

Outside Discussions

	NEG	PART	POS	UNC
ABBA	2		41	
ABBX				
ABBC				
AXBX			1	
AXBA			15	
AXBC				
ABCA	1		8	1
ABCB				
ABCD			1	
ABCX			1	
AXAX		1	14	
ABAX			1	
ABAB				
AXAB				
ABAC				

IS1003c

Not Available.

IS1003d

H

Inside Discussions

	NEG	PART	POS	UNC
ABBA	5		35	2
ABBX	4	1	5	
ABBC			1	
AXBX	1		24	
AXBA	8		34	4
AXBC			3	1
ABCA	1	1	6	1
ABCB				

ABCD			4	1
ABCX	1		5	1
AXAX	1		4	1
ABAX	1		1	
ABAB				
AXAB				
ABAC				

Outside Discussions

	NEG	PART	POS	UNC
ABBA	2	2	17	
ABBX				
ABBC			1	
AXBX	1		8	2
AXBA	2	3	27	3
AXBC				1
ABCA		1	5	
ABCB				
ABCD			3	
ABCX		1	5	1
AXAX			1	
ABAX				
ABAB				
AXAB				
ABAC				

J

Inside Discussions

	NEG	PART	POS	UNC
ABBA	7	2	42	2
ABBX	4	1	3	
ABBC			2	
AXBX	3		26	1
AXBA	8	2	51	5
AXBC			2	1
ABCA	1	2	7	1
ABCB				
ABCD			3	1
ABCX	1		10	1
AXAX	1		4	1
ABAX	1		1	
ABAB				
AXAB				
ABAC				

Outside Discussions

	NEG	PART	POS	UNC
ABBA			10	
ABBX			2	
ABBC				
AXBX			6	1
AXBA	2	1	10	2
AXBC			1	1
ABCA			4	
ABCB				
ABCD			4	
ABCX		1		1
AXAX			1	
ABAX				
ABAB				
AXAB				
ABAC				

R

Inside Discussions

	NEG	PART	POS	UNC
ABBA	4		10	1
ABBX	4		1	
ABBC			2	
AXBX	1		10	
AXBA	3		13	1
AXBC			1	
ABCA		1	1	
ABCB				
ABCD				1
ABCX	1		4	
AXAX			2	
ABAX	1		1	
ABAB				
AXAB				
ABAC				

Outside Discussions

	NEG	PART	POS	UNC
ABBA	3	2	42	1
ABBX		1	4	
ABBC			2	
AXBX	2		22	2
AXBA	7	3	48	6
AXBC			2	2
ABCA	1	1	10	1
ABCB				

```
ABCD              7
ABCX        1     6     2
AXAX  1           3     1
ABAX
ABAB
AXAB
ABAC
```

APPENDIX E

TAS

TAS or Twente Argument Schema is a method if visualising and structuring discussions based on nodes and relations between those nodes. An example of such a visualisation is given in figure E.1. In TAS an utterance is assigned a label from a fixed set of labels. Back-channels and related sentences are filtered out by giving them the "Unknown" label [Ver06]. A detailed description of TAS is available in [vdW05] and in [Ver06]. The node-labels can be subdivided into statements and issues. The following labels may be assigned to nodes.

Statement A node is labelled as a Statement when that node constitutes a claim.

Weak Statement A "Weak Statement" is a statement with little force behind the claim, the person stating it is not sure.

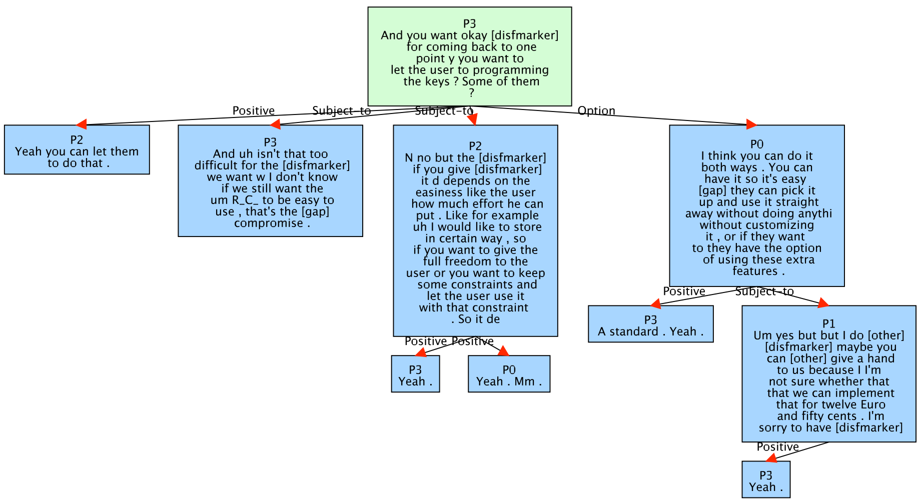
Open Issue From [vdW05]: "This label is to label issues that are raised where every possible response could be a solution. 'Wh. questions ' *Example* :'What does it prove?' can be labelled as Open Issue."

A/B Issue Nodes are Labelled with 'A/B Issue' when it is a question where the possible responses are explicitly enumerated.

Yes/No Issue Nodes are Labelled with 'Yes/No Issue' when issues are raised where the possible responces are yes and no.

Unknown Nodes labelled with unknown are not shown in the diagrams. The unknow tag is used to label backchannel utterances. These are not show because they do not add to the understanding of the discussion.

As visible in figure E.1 TAS structures the nodes based on relations between those nodes. Each Node, except the root node, has a relation to a parent node. A node becomes a child of the previous node, unless the parent of the parent is more apropriate (all the way up to the root node)[RR06b]. TAS contains several relation labels:


P3
And you want okay [disfmarker] for coming back to one point y you want to let the user to programming the keys ? Some of them ?

P2
Yeah you can let them to do that .

P3
And uh isn't that too difficult for the [disfmarker] we want w I don't know if we still want the um R_C_ to be easy to use , that's the [gap] compromise .

P2
N no but the [disfmarker] if you give [disfmarker] it d depends on the easiness like the user how much effort he can put . Like for example uh I would like to store in certain way , so if you want to give the full freedom to the user or you want to keep some constraints and let the user use it with that constraint . So it de

P3
Yeah .

P0
Yeah . Mm .

P0
I think you can do it both ways . You can have it so it's easy [gap] they can pick it up and use it straight away without doing anythi without customizing it , or if they want to they have the option of using these extra features .

P3
A standard . Yeah .

P1
Um yes but but I do [other] [disfmarker] maybe you can [other] give a hand to us because I I'm not sure whether that that we can implement that for twelve Euro and fifty cents . I'm sorry to have [disfmarker]

P3
Yeah .

Figure E.1: *An example of a discussion visualised using TAS. For this a discussion from meeting IS1003c from the AMI corpus was used.*

Positive Supports the parent.

Negative Child means to refute the parent.

Uncertain Contribution for the participant is unclear [RR06b]. This means that it is unclear wether the contribution is positive or negative towards the parent.

Option The child is a possible solution to the parent. Example (as used in [vdW05]): *The relation between the open issue, 'What is the capital of Moldova?' and the statement 'I would say Chisinau' can be labelled as an option relation.*

Option Exclusion The child excludes one or more options, solutions or answers from the parents statements.

Elaboration The child asks the parent for clarification, elaboration *or* the child repeats the parent with a different wording.

Specialization A specalisation can be applied when a particular issue generalises or specialises another issue.

Subject to The child points out critera or dependencies related to the parent, these will have to be evaluated before the parent can be supported or refuted.

E.1 Remarks

There seem to be some parralels between TAS and the AMI DA's. At least as far as the elicit- DA labels is concerned, those seem to correspond with the TAS

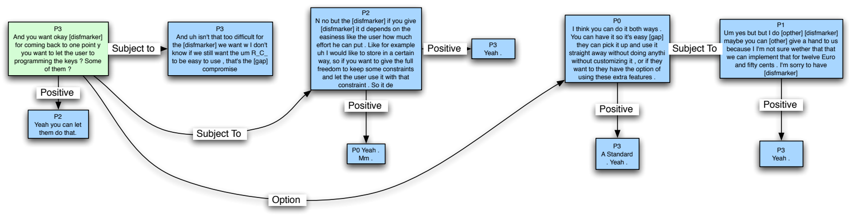

P3
And you want okay [disfmarker] for coming back to one point y you want to let the user to programming the keys ? Some of them ?
Subject to
P3
And uh isn't that too difficult for the [disfmarker] we want w I don't know if we still want the um R_C_ to be easy to use , that's the [gap] compromise
P2
N no but the [disfmarker] if you give [disfmarker] it d depends on the easiness like the user how much effort he can put . Like for example uh I would like to store in a certain way, so if you want to give the full freedom to keep some constraints and let the user use it with that constraint . So it de
Positive
P3
Yeah .
P0
I think you can do it both ways . You can have it so it's easy [gap] they can pick it up and use it straight away without doing anythi without customizing it , or if they want to they have the option of using these extra features .
Subject To
P1
Um yes but but I do [opther] [disfmarker] maybe you can [other] give a hand to us because I I'm not sure wether that that we can implement that for twelve Euro and fifty cents . I'm sorry to have [disfmarker]
Positive
P2
Yeah you can let them do that.
Subject To
Positive
P0 Yeah . Mm .
Positive
P3
A Standard . Yeah .
Positive
P3
Yeah .
Option

Figure E.2: *An alternative visualisation from the one provided in figure E.1. The same discussion is used as example. This option should be read from left to right, from top to bottom.*

issue labels. A lot of other DA labels (inform, suggest, offer, asses) correspond to TAS statements. And finally the DA-labels backchannel, stall, fragment and other can be directly mapped onto the TAS 'unknown' label.

In AMI DA's are complemented by a small set of relation labels, namely *positive, negative, partial, uncertain*. These relations are only annotated for cases where there actually is a clear relation. On the other hand, TAS has a high number of relation labels, and all its nodes have a relation to some parent, except the first node of a discussion. It is important to note that the AMI "unknown" relation is different than the TAS one. It is unclear what TAS adds to the DA-annotation when that annotation is augmented with the small set of relations as is done in the AMI corpus, however it seems to work. And TAS is too different to make any direct comparisons.

There are some unclear things in the TAS documentation. For one disambiguation is not profided where it is needed. For example the phrase "but isn't it so that . . . ?" can be a yes/no issue as well as a statement. The TAS documentation and the work on TAS also omits to define what a discussion is, we did some preliminary work in chapter 2.

TAS has good points as well. Verbree shows in [Ver06] that classification of utterances in TAS is possible. He made some headway and achieved a best result of 78.52% on the classification of nodes and he outlines an approach and statistics for classifying the relations as well. Also using the TAS representation for discussions helps people comprehend these discussions faster, as shown in [RR06a]. However during the thesis a few alternatives to this visualisation came to mind. These can be seen in figure E.2 for a left-right perspective and in figure E.3 for the top-bottom perspective.

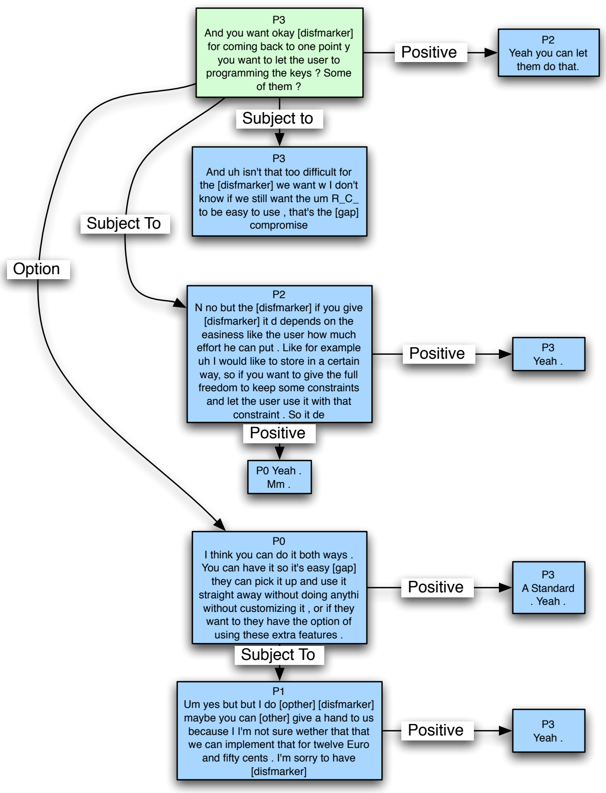

P3
And you want okay [disfmarker] for coming back to one point y you want to let the user to programming the keys ? Some of them ?
Positive
P2
Yeah you can let them do that.
Subject to
P3
And uh isn't that too difficult for the [disfmarker] we want w I don't know if we still want the um R_C_ to be easy to use , that's the [gap] compromise
Subject To
Option
P2
N no but the [disfmarker] if you give [disfmarker] it d depends on the easiness like the user how much effort he can put . Like for example uh I would like to store in a certain way, so if you want to give the full freedom to keep some constraints and let the user use it with that constraint . So it de
Positive
P3
Yeah .
Positive
P0 Yeah .
Mm .
P0
I think you can do it both ways . You can have it so it's easy [gap] they can pick it up and use it straight away without doing anythi without customizing it , or if they want to they have the option of using these extra features .
Positive
P3
A Standard
. Yeah .
Subject To
P1
Um yes but but I do [opther] [disfmarker] maybe you can [other] give a hand to us because I I'm not sure wether that that we can implement that for twelve Euro and fifty cents . I'm sorry to have [disfmarker]
Positive
P3
Yeah .

Figure E.3: *An alternative visualisation from the one provided in figure E.1. The same discussion is used as example. This option should be read from top to bottom, from left to right.*

APPENDIX F

Selected Cue Phrases

In this appendix we will list 200 cue-phrases selected by the DCP cue-phrase selection method. In the first section for the Addressee determination task (single or group addressed) on the filtered corpus. In the second section for the addressee determination task (single or group addressed) on the corpus without any unknown sentences. In the third section for the recipe classification task. The cue-phrase lists also contain what class the phrase is a cue-phrase for.

F.1 Addressee

and	--- G
to	--- G
have	--- G
uh	--- G
That's	--- U
um	--- G
then	--- G
this	--- G
I don't	--- U
that	--- G
But	--- U
No	--- S
will	--- G
there	--- U
remote	--- G
all	--- G
see	--- G
don't	--- U
we	--- G
which	--- G
to take care	--- U
control	--- G
take care of	--- U
I'm a	--- U
take care	--- U
care	--- U
what	--- G
think so .	--- U
great .	--- U
makes	--- U
thought	--- U
And you	--- U
about	--- G
of	--- G
Um	--- G
say	--- U
Very	--- U
to take	--- U
drawing	--- U
great	--- U
Well ,	--- U
very	--- G

anything	--- U
think so	--- U
use	--- G
buttons .	--- U
name	--- U
some	--- G
Not	--- U
one	--- G
should	--- G
in	--- G
Ah	--- U
a	--- G
are	--- G
how	--- G
want	--- G
button	--- G
would be	--- U
has	--- G
our	--- G
from	--- G
Oh ,	--- U
something	--- G
, yeah .	--- S
at	--- G
off	--- G
now	--- G
might	--- U
controls	--- G
on the	--- G
T_V_	--- G
Let's	--- U
come	--- G
Uh ,	--- U
like	--- G
w	--- U
its the	--- U
generation if	--- U
are here for	--- U
recommendations	--- U
sell oils with	--- U
people anyway	--- U
to simply we'll	--- U
It's number	--- U
standard .	--- U
site	--- U
the new remote	--- U
silly	--- U
Russian trick	--- U
Any	--- U
flops for	--- U
celebration	--- U
simply we'll	--- U
have um have	--- U
out of wood	--- U
be made out	--- U
, like Amazon	--- U
, for the	--- U
Christine can	--- U
design and working	--- U
, that's expensive	--- U
of the control	--- U
me try	--- U
have time to	--- U
, I would	--- U
a PowerPoint presentation	--- U
another really good	--- U
we win	--- U
We we	--- U
you could sell	--- U
uh I'm not	--- U
the first generation	--- U
gets lost	--- U
And our	--- U
design meeting mm	--- U
another really	--- U
the wrong folder	--- U
could sell	--- U
this model	--- U
Few buttons	--- U
th of the	--- U
So , you	--- U
that we haven't	--- U
I know this	--- U
characteristic	--- U
be to	--- U
the o	--- U
requirement specification ,	--- U
lets me	--- U
technical function design	--- U
the depending on	--- U
. That	--- U
configure	--- U
it's channel up	--- U
the orangutan	--- U
, because people	--- U
sports time	--- U
I mean .	--- U
Mushroom	--- U
made out of	--- U

```
Alright ,                     --- U
stay                          --- U
artist                        --- U
imply good                    --- U
point of view                 --- U
I’m missing                   --- U
Okay uh tha                   --- U
of view                       --- U
be cost                       --- U
reason that we                --- U
doesn’t .                     --- U
the wrong                     --- U
, technical                   --- U
Sammy Benjo                   --- U
if you go                     --- U
of that point                 --- U
recorded ? Okay               --- U
Designer , okay               --- U
o th                          --- U
sheep                         --- U
have time                     --- U
uh Yeah                       --- U
finishing we                  --- U
of .                          --- U
the for the                   --- U
uh team                       --- U
channel up channel            --- U
So before                     --- U
uh tha                        --- U
far .                         --- U
mood , or                     --- U
that’s another                --- U
do the action                 --- U
with it ,                     --- U
No it doesn’t                 --- U
she wanted                    --- U
imagine .                     --- U
it’s nice ,                   --- U
Welcome back everybody        --- U
yeah I think                  --- U
the for                       --- U
performance .                 --- U
And it can                    --- U
Okay , electronics            --- U
uh what would                 --- U
can uh define                 --- U
Few                           --- U
win                           --- U
display also                  --- U
the depending                 --- U
good for                      --- U
bit short                     --- U
direction ,                   --- U
know , things                 --- U
have to work                  --- U
depending on your             --- U
presentation , good           --- U
is not often                  --- U
Okay well                     --- U
```

F.2 Addressee Without Unknown

```
 to                           --- G
the                           --- G
we                            --- G
and                           --- G
uh                            --- G
Yeah                          --- S
have                          --- G
I                             --- G
a                             --- G
?                             --- S
this                          --- G
that                          --- G
is                            --- G
of                            --- G
.                             --- S
for                           --- G
will                          --- G
remote                        --- G
,                             --- G
are                           --- G
it                            --- G
be                            --- G
um                            --- G
on                            --- G
So                            --- G
some                          --- G
then                          --- G
can                           --- G
our                           --- G
yeah                          --- S
more                          --- G
in                            --- G
```

all	--- G
very	--- G
so	--- G
control	--- G
what	--- G
like	--- G
with	--- G
And	--- G
if	--- G
use	--- G
just	--- G
No	--- S
Okay	--- S
they	--- G
also	--- G
one	--- G
think	--- G
maybe	--- G
design	--- G
Yes	--- S
controls	--- G
should	--- G
about	--- G
which	--- G
It’s	--- S
want	--- G
know	--- G
new	--- G
okay	--- S
off	--- G
first	--- G
because	--- G
but	--- G
You	--- S
little	--- G
simple	--- G
We	--- G
them	--- G
things	--- G
easy	--- G
T_V_	--- G
buttons	--- G
now	--- G
meeting	--- G
would	--- G
from	--- G
see	--- G
could	--- G
device	--- G
these	--- G
Um	--- G
good	--- G
do	--- G
you	--- G
there	--- G
important	--- G
Mm-hmm	--- S
something	--- G
don’t	--- G
out	--- G
it’s	--- G
product	--- G
people	--- G
do you	--- S
not	--- G
s	--- G
make	--- G
you’re	--- S
how	--- G
wheel	--- G
w	--- G
project	--- G
thing	--- G
other	--- G
into	--- G
come	--- G
might	--- G
has	--- G
here	--- G
need	--- G
user	--- G
sort	--- G
when	--- G
most	--- G
quite	--- G
no	--- S
Oh	--- S
get	--- G
we’re	--- G
take	--- G
case	--- G
going	--- G
Uh	--- S
fifty	--- G
find	--- G
there’s	--- G
fancy	--- G
actually	--- G
you think	--- S

different	--- G
remotes	--- G
next slide	--- S
their	--- G
at	--- G
button	--- G
main	--- G
right	--- S
idea	--- G
as	--- G
Euro	--- G
presentation	--- G
functions	--- G
discuss	--- G
time	--- G
say	--- G
yes	--- S
an	--- G
bit	--- G
look	--- G
each	--- G
but uh	--- S
those	--- G
send	--- G
powerful	--- S
gonna	--- G
slide	--- S
requirements	--- S
next	--- S
least	--- G
minutes	--- G
price	--- G
colour	--- G
we've	--- G
didn't	--- G
feel	--- G
where	--- G
used	--- G
Like	--- S
speech	--- G
Of course	--- S
sell	--- G
let's	--- G
one button	--- S
Sure	--- S
marketing	--- S
your	--- S
or	--- G
What	--- S
choose	--- G
were	--- G
individual	--- G
components	--- G
decision	--- G
channel	--- G
both	--- G
please	--- S
'cause	--- G
push	--- G
well	--- G
really	--- G
my	--- G
point	--- G
why	--- G
interface	--- G
Designer	--- S
fine	--- S
too	--- G
Yep	--- S

F.3 Recipe

Lekker met	--- suggestion
echt	--- explanation
Lekker	--- suggestion
niet	--- explanation
g	--- inf-relevant
ik	--- explanation
1	--- requirement
Bereiding:	--- signal
personen	--- inf-relevant
Eet	--- other
minuten	--- ins-nonblock
en laat	--- ins-nonblock
maar	--- explanation
dus	--- emphasis
Ingredinten:	--- signal
	--- inf-relevant
gaar.	--- explanation
dat de	--- emphasis
Bron	--- inf-irrelevant
bron	--- inf-irrelevant
kookboek.nl	--- inf-irrelevant
gram	--- requirement

```
wijnen                        --- suggestion
smaken                        --- inf-irrelevant
schoon uitkomt, is            --- explanation
veel handelingen              --- explanation
grote in                      --- explanation
pen                           --- explanation
de geur van                   --- explanation
tikkie meer tijd,             --- explanation
zaden                         --- explanation
maakt het                     --- explanation
makkelijk en snel             --- explanation
volgt zelf te                 --- explanation
volgens een                   --- explanation
is als                        --- explanation
verband met de                --- explanation
er gewoon aan,                --- explanation
het toch een                  --- explanation
de geur                       --- explanation
verband met                   --- explanation
klaargemaakt.                 --- explanation
schoon uitkomt,               --- explanation
kruiden in het                --- explanation
anders nl.                    --- explanation
het er volgens                --- explanation
een recept vanaf              --- explanation
snel gerecht.                 --- explanation
die zul je                    --- explanation
Tomatenolie                   --- explanation
Dit doe je                    --- explanation
handelingen en                --- explanation
Met 1/4 liter                 --- explanation
je goed                       --- explanation
handelingen                   --- explanation
Omdat                         --- explanation
als het er                    --- explanation
1/4 liter zit                 --- explanation
als volgt                     --- explanation
en is                         --- explanation
Scheelt je                    --- explanation
volgt zelf                    --- explanation
heel smerig bij               --- explanation
vanaf moet -                  --- explanation
is de taart                   --- explanation
doe je om                     --- explanation
volgt                         --- explanation
in het deeg                   --- explanation
ging het er                   --- explanation
een tikkie meer               --- explanation
Met 1/4                       --- explanation
liter zit je                  --- explanation
toch een makkelijk            --- explanation
meer tijd,                    --- explanation
- die                         --- explanation
kleffe zooi                   --- explanation
ging                          --- explanation
uitkomt, is de                --- explanation
een recept                    --- explanation
overal                        --- explanation
zul                           --- explanation
nl.                           --- explanation
velletje er gewoon            --- explanation
garingstijd.                  --- explanation
gelijke grote                 --- explanation
ook als het                   --- explanation
uitkomt,                      --- explanation
goed -                        --- explanation
werk!                         --- explanation
hebben.                       --- explanation
je om                         --- explanation
zooi                          --- explanation
deeg te                       --- explanation
smerig                        --- explanation
de tomatensmaak en            --- explanation
allemaal nodig                --- explanation
geur van de                   --- explanation
Kost                          --- explanation
een tikkie                    --- explanation
geur van                      --- explanation
velletje er                   --- explanation
maken:                        --- explanation
dat maakt                     --- explanation
vocht uit.                    --- explanation
het zijn                      --- explanation
er gewoon                     --- explanation
(Meestal                      --- explanation
tomatensmaak en               --- explanation
makkelijk                     --- explanation
Liefst van                    --- explanation
aan, ook                      --- explanation
van de kruiden                --- explanation
tomatensmaak                  --- explanation
liter zit                     --- explanation
Ze worden anders              --- explanation
hangen.                       --- explanation
geur                          --- explanation
tijd,                         --- explanation
overigens                     --- explanation
gewoon aan,                   --- explanation
Liefst                        --- explanation
```

kleffe	--- explanation
moet -	--- explanation
worden anders	--- explanation
zelf te	--- explanation
tikkie meer	--- explanation
de kruiden in	--- explanation
er volgens	--- explanation
tikkie	--- explanation
je een hoop	--- explanation
versterkt	--- explanation
van gelijke	--- explanation
recept vanaf	--- explanation
en dat	--- explanation
in verband	--- explanation
Ze worden	--- explanation
een hoop	--- explanation
het toch	--- explanation
er schoon	--- explanation
zit je	--- explanation
Dit doe	--- explanation
het er	--- explanation
wil verwijder	--- explanation
maakt	--- explanation
gerecht.	--- explanation
Scheelt	--- explanation
ook als	--- explanation
te laten	--- explanation
hoop	--- explanation
toch een	--- explanation
met wijn	--- explanation
vanaf moet	--- explanation
je een	--- explanation
toch	--- explanation
veel	--- explanation
en het vocht	--- explanation
verband	--- explanation
kruiden in	--- explanation
meer	--- explanation
allemaal	--- explanation
1/4 liter	--- explanation
gelijke	--- explanation
laten trekken.	--- explanation
zit	--- explanation
om de	--- explanation
verwijder	--- explanation
tot	--- ins-nonblock
aan,	--- explanation
als het	--- explanation
het velletje	--- explanation
zachtjes	--- ins-nonblock
velletje	--- explanation
met bier	--- explanation
4	--- inf-relevant
Benodigdheden	--- signal
uit.	--- explanation
1/4	--- explanation
witte	--- suggestion
Laat	--- ins-nonblock
Eventueel	--- suggestion
trekken.	--- explanation
schoon	--- explanation
Ingredinten	--- signal
Smakelijk	--- other
(vrijwel)	--- other
smakelijk!	--- other
nog steeds	--- other
Niet	--- emphasis
Let	--- emphasis
eten!	--- other
steeds	--- other
wel	--- emphasis
bij de	--- other
mousse	--- emphasis
anders	--- explanation
Bereidingstijd:	--- inf-relevant
het vocht	--- explanation
doe je	--- explanation
gerecht is	--- inf-irrelevant
volgens	--- explanation