On the Structuring of Discussion Transcripts Based on Utterances Automatically Classified

A.T. Verbree
Master of Science Thesis
Computer Science

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 (first supervisor)

may 2006

Summary

This report describes the development of a method of creating argument diagrams of discussions. An argument structure called the Twente Argumenta Schema (TAS) is described together with a corpus of argument diagrams of discussions. Methods are presented to extract values out of several characteristics of utterances in discussions, most of them concerning combinations of words and part of speech tags. Using these values experiments are performed on the classification of utterances in discussions according to TAS. We present several classification results, using as well different values for parameters as different features. Our best performance of 78.52% is a nice improvement over the baseline of classifying utterances according to TAS. Still many improvements can be made, especially in the selection of the right word combinations as characteristics of a class of arguments.

Our work has its focus on the classification of utterances in discussions according to TAS, but we also present an approach on the identification and classification of the relations between utterances.

Since our TAS classification shows much resemblance with Dialog Act Tagging our method has as well been used for Dialog Act Tagging, with good results.

Samenvatting (Summary in Dutch)

Dit verslag beschrijft de ontwikkeling van een methode om vanuit discussies argumentatie diagrammen te creëeren. Een argumentatie structuur genaamd het Twente Argumentatie Schema (TAS) is beschreven tezamen met een corpus van argumentatie diagrammen van discussies. Methodes om waarden te verkrijgen uit de verschillende karakteristieken van een uiting in discussies, waarvan de meeste aangaande combinaties van woorden van woordsoorten, worden gepresenteerd. Gebruik maken van deze eigenschap-waarden zijn experimenten uitgevoerd om uitingen in discussies te kunnen classificeren naar TAS. We presenteren classificatieresultaten door verschillende waarden te geven aan parameters alsmede door gebruik te maken van verschillende eigenschappen van uitingen. Ons beste resultaat van 78.52% is een aardige verbetering over de *'baseline'* van het classificeren van uitingen op basis van TAS. Desalniettemin kunnen er nog veel verbeteringen aangebracht worden, met name betreffende de selectie van de juist woord combinaties als karakteristieken voor een klasse van argumenten.

Ons werk heeft zich gefocused op de classificatie van uitingen in discussies gebruik makend van TAS, maar we presetenreren ook reeds een benadering om relaties tussen uitingen te identificeren en te classificeren.

Aangezien onze TAS classificatie erg veel gelijkenis vertoond met 'Dialog Act Tagging', beiden richten zich namelijk op het classificeren van uitingen, hebben we onze methode ook ingezet bij het 'Dialog Act Tagging', met goede resultaten als gevolg.

Preface

(Dutch:)
'Ik heb een uitnodiging gehad voor een congres,' vertelde hij toen ze aan tafel zaten.
'Alweer?' vroeg ze
'Alweer?' - haar reactie irriteerde hem. 'Ik krijg bijna nooit een uitnodiging.'
'En je bent net naar een congres geweest!' - haar stem was verontwaardigd. 'Je doet niet anders meer als naar congressen gaan! Je lijkt Beerta wel.'
Hij dwong zichzelf tot kalmte. 'Dat was drie jaar geleden.'
'Twee jaar!'
'Drie jaar!'
'Twee jaar!'
In ieder geval heeft dat er niets mee te maken. Dit is een heel ander congres.'
'Wat is het dan voor congres?'
'Dit is een feestcongres.'
'Een feestcongres?' - haar stem ging omhoog van verontwaardiging, 'Maar daar hoef je toch zeker niet naar toe? Wat is dat, een feestcongres?'
'De Belgische Commissie bestaat vijfentwintig jaar.'
'En moet jij daar naar toe? Daar heb je toch zeker niets mee te maken? Je lacht toch zeker om zulke onzin? Toen jullie vijfentwintig jaar bestonden hebben jullie toch ook geen feestcongres gegeven?'
'Nee,' gaf hij toe.

(English:)
'I've had an invitation for a congress,' he told when they sat at the table.
'Again?' she asked
'Again?' - her reaction irritated him. 'I hardly ever get an invitation.'
'And you've just been to a congress!' - her voice was incensed. 'All you do is going to congresses! You seem like Beerta.'
He forced himself to stay calm.'That was three years ago.'
'Two years!'
'Three years!'
'Two years!'
'Anyway it doesn't have anything to do with this. This is a complete different congress.'
'So, what kind of congress is it then?'
'It is a celebration-congress.'
'A celebration-congress?' - her voice rose of incense, 'but you don't have to go there do you? What is it, a celebration-congress?'
'The Belgium committee exists for twenty five years.'
'And you have to go there? But you don't have anything to do with that, do you? You always laugh about such nonsense? When you're bureau existed for twenty five years

you didn't organize a celebration-congress, did you?'
'No,' he admitted.

'Het bureau' (The Bureau), by J.J. Voskuil.

By finishing my master of science thesis I can look back on seven years of study in computer science. Although there almost always seemed to be even more interesting things than computer science I now have to admit that I enjoyed studying it, especially the last years of it.
At first, when I started my project on discovering meeting structures I had big ideas like implementing and using it in meetings of the VGST, the christian student association I'm involved in. I hoped to make each meeting a meeting of celebration, a party by it self. But the further I came in the project and the more time I spend I discovered that once again I was a little naive. Not only does science invent the future, but also does it need the future to keep inventing things: good research takes a lot of time.
Having spend 12 months on this project I am thankful to a lot of people who kept me motivated by asking the right questions or by refraining from asking questions at the right time. Just as explaining my project to others kept me motivated getting better performances and thus developing a better classifying system, not talking about my project but turning my mind to other things helped me to start (almost) each day with a happy feeling: so much to discover and do today!
Thanks to Rutger, my first supervisor who checked on me regularly and who showed himself to be very involved in my work. Thanks as well to my other two supervisors: Rieks and Dirk who helped me with their comments and gave me interesting insights and tips. But not only my supervisors need to be thanked, a lot of other people who were interested in my project as well, such as my parents, other family and friends.
Special thanks to my (former) flatmates: Rudolf, Werner, Jan-Maarten and Niek always were great motivators, although sometimes the evenings were too much fun, so I had to start later the day after.
Having thanked so many people I want to thank Ellis in special. Every time when I spoke about my project she listened, whether I was positive or not, she always was willing to help me. By giving me feedback and support but even more important the right distraction she kept me motivated for finishing my masters.

Daan Verbree
Enschede, May 2006

Contents

Chapter 1

Introduction

1.1 Introduction

Nowadays having a job almost means having to do with a lot of meetings. In all these meetings about everything can be discussed: From how much cups of coffee should be compensated for by the employer to the introduction of a new coffee machine. But meeting time is not only spend during the meeting, it is spend between meetings as well: Minutes have to be made and read and action items have to be distilled from them.

These ongoing meetings have been a point of interest to many researchers. Psychologists like to know what people experience in meetings and why they act like they do, physiotherapists might be interested in the posture of persons attending a meeting and computer scientists are interested in the way they might aid a meeting to improve efficacy. An example of the last is in the work of Ellis et al. [2001] which researches the possibilities of the use of computer agents in meetings. Three different sorts of agents are presented: information, social and organizational agents. Information agents might help in gathering and presenting information to all the participants, social agents might keep track of the time a meeting takes and propose coffee breaks at the right time and finally organizational agents could remind all participants of approaching deadlines, summing up all action items and making the minutes instantly available.

Another example of ongoing research in meetings is a European project called Augmented Multi-party Interaction (AMI) which has started a few years ago, researching new technologies to support human interaction, in the context of smart meeting rooms and remote meeting assistants. Just as in Ellis' work these remote meeting assistants could assist in taking meeting minutes or signal the chairman when a participant tends to get too dominant.

Both of these project realize that perhaps one of the most time consuming activities involving meetings is (except from the meeting it self) taking minutes and analyzing them for action points and other points of (personal) interest.

Alongside the increasing amount of meetings another interesting development is ongoing: virtual conferencing. By virtual conferencing we mean having meetings in a virtual meeting room, a sort of highly developed chatbox. In a virtual meeting room participants can be anywhere in the world but still have a meeting, by logging on to the room. It is highly likely that in a virtual meeting room the chairman has more influence

than other participants and therefore a virtual chairman as described by Rienks et al. [2005b] could be an interesting meeting assistant.

To construct a virtual chairman which is able to lead a meeting all by itself, giving turns, keeping track of a time-line and most important: keeping the meeting as effective and efficient as possible a lot of research is needed. A virtual chairman should not only be aware of the *state* of the participants as their dominance level, which is researched in [Rienks et al., 2005a, Rienks and Heylen, 2006], but a trustworthy representation of the meeting might even be more important.

The point where this trustworthy representation of a meeting meets the minute making and analyzing as described above is in our project. Our work concerns the *automatic* constructing of argument diagrams of discussions. Not only can these diagrams be used to represent meetings in a computational way, but arguments diagrams themselves also represent a discussion in such a way that it leads to quicker cognitive comprehension, deeper understanding and enhanced detection of weaknesses [Schum and Martin, 1982, Kanselaar et al., 2003]. Furthermore they are said to aid the decision making process, and can be used as an interface for communication to maintain focus, prevent redundant information and to save time [Yoshimi, 2004, Veerman, 2000].

In our work we have focused on the research and development of models of utterances in order to classify utterances according to the models developed. A model of such an utterance is a list of its' features and the values for these features. An example a model of an utterance is the length, measured in the number of words $Length(Utterance) = 7$. The models and methods developed have been applied in the classification of utterances in AMI discussions, classified according to the Twente Argument Schema (TAS). But not only have these methods been applied to classify utterances according to TAS, they have also been evaluated on other annotated corpora, such as the ICSI meeting corpus, thus being able to compare our models and methods to earlier performances in the accuracy of the classification as described in the literature. Not only is TAS used to classify utterances, but as well binary relations between nodes. We present an approach to this identification and classification

In this master thesis we will first introduce the Twente Argument Schema together with the HUB corpus in chapter 2, followed by a description of the features we use to classify utterances in discussions according to TAS. Chapters 4, 5 and 6 will handle our classification experiments performed and the results obtained using different classification techniques. A comparison between our TAS classification and Dialog Act classification as done by Ang et al. [2005] and many others is made in chapter 8. We conclude our thesis with conclusions and recommendations for further research.

Chapter 2

TAS

2.1 Introduction

Our work concerns automatic creating of argument diagrams of discussions or to phrasing it a bit different: The learnability of structuring discussions according to the Twente Argument Schema (TAS), which has been developed at the University of Twente [Weijden, 2005]. TAS has been developed while working on the creation of a new corpus, called the AMI Hub Corpus. The AMI Hub Corpus contains a total of 80 meetings into 20 series of 4 meetings. These meetings have been recorded at the University of Edinburgh (United Kingdom), the IDIAP research Institute (Switzerland) and at TNO (the Netherlands) [Carletta et al., 2005]. For these meetings video, sound and transcriptions are available. The corpus is still under development and several annotation layers as Dialog Acts, gestures, focus of attention and topic information are being created. Transcriptions were created for all the meetings in the AMI corpus, following strict annotation guidelines [Moore et al., 2005]. For our research we will only make use of these transcriptions of the HUB corpus. This chapter will give a short introduction to TAS, annotating using TAS and agreement amongst annotators.

2.2 Annotation

TAS is not the only scheme available for structuring argumentative texts, several other argument structure methods have been developed before. Weijden [2005] reports on a research of argument structure methods as Rhetorical Structure Theory (RST), Toulmin and Issue-Based Information Systems (IBIS). Although each of the structure methods has there own advantages, none of them seems to be fit to structure the discussions and argumentation found in a meeting. This analysis resulted in the development of the Twente Argument Schema, which is partly based on other argument structures.

TAS is a structuring method based on nodes and arcs. For each utterance in a discussion a label can be picked from a fixed list which describes the intention of the utterance best. By assigning a label to an utterance we have created a node. Arcs can be created by creating relations between two nodes and labeling this relation with a type which as well can be picked from a fixed list. Restrictions are made on the assigning of relations between nodes, based on the type of the nodes. Table 2.1 shows the labels available in TAS for as well utterances as relations. In the next paragraphs we will give an overview of these nodes and relations, more extensive descriptions of these labels

can be found in A. An example of a generated scheme by applying TAS is presented in figure 2.1.

Utterance labels	*Relation labels*
Statement	Positive
Weak Statement	Negative
Open Issue	Uncertain
A/B Issue	Option
Y/N Issue	Option-exclusion
Unknown	Elaboration
	Specialization
	Subject-to

Table 2.1: Labels for utterances and relations in TAS

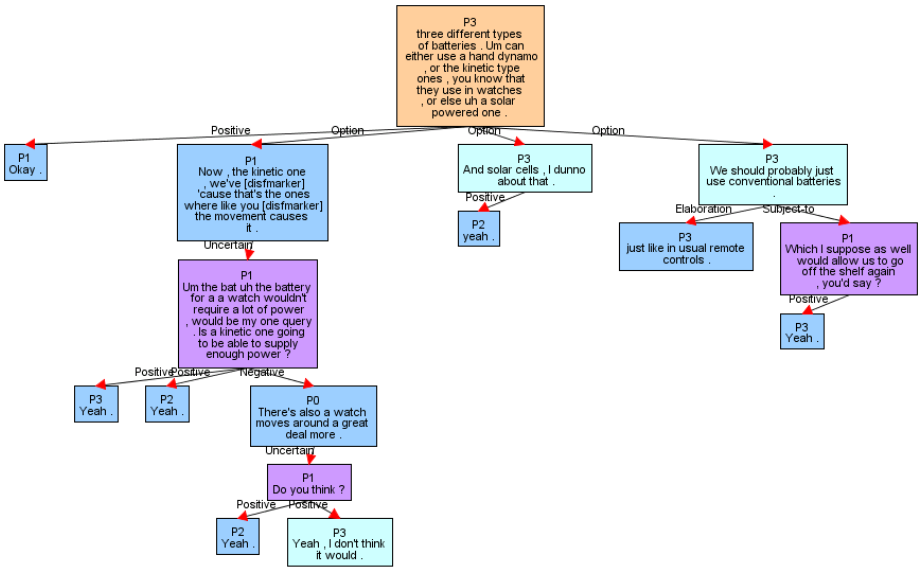

Figure 2.1: TAS diagram example

2.2.1 The Node Labels

In line with Galley et al. [2004] backchannel utterances such as “uhhuh” and “okay” are filtered out and to be neglected, since they are generally used by listeners to indicate they are following along, and not necessarily indicating (dis)agreement. These utterances are classified as *unknown* but are not shown in diagrams since they do not add any information. The nodes in our argument diagrams consist of issues and statements.

In the Twente Argument Schema three different labels for nodes to represent issues have been defined: The *open issue*, the *a/b issue* and the *yes/no issue*. The *open issue* allows for any number of possible replies possibly revealing positions or options that were not considered beforehand (c.f “What what do you think , Craig ?”). This in

contrast with the *a/b issue*, that allows participants to take a position for a countable number of positions which should be known from the context (c.f. "Would you say ants, cats or cows?"). The *yes/no issue*, in line with the yes-no question in IBIS [Kunz and Rittel, 1970] directly requests whether the participants positions agree or disagree with the issue (c.f."Do we need an LCD display ?").

Participants' positions are generally conveyed through the assertion of a *statement*. The content of a *statement* always contains a proposition in which a certain property or quality is ascribed to a person or thing. A proposition can be a description of facts or events, a prediction, a judgement, or an advice [Van Eemeren et al., 2002] (c.f. "Okay , so fifteen to thirty five , look fairly young . You know , they have bit of expendable income to spend on this sort of thing .") . Statements can vary in the degree of force and scope. It can happen that meeting participants make remarks that indicate that they are not sure if what they say is actually true. Toulmin [1958] uses a qualifier in his model to say something about the force of what he calls 'claim'. When this qualifier is introduced, it is possible that the assertion is made with less force. As Eemeren [2003] points out that the force of an argument can also be derived from lexical cues such as by expressing the words 'likely' and 'probably'. To be able to represent this we introduce the label *'weak statement'* (c.f. "Um I guess that's up to us , I mean you probably want some kind of unique selling point of it , so um , you know").

2.2.2 The Relation Labels

Relations can only exist between nodes. For this we have defined a number of relations that can exist between the labelled nodes. When engaged in a discussion or debate, the elimination of misunderstandings is a prerequisite in order to understand each other and hence to proceed [Neass, 1966]. Participants in a discussion, according to Neass, eliminate misunderstandings by clarifying, or specifying their statements. These moves can e.g. be observed in the criteria definition phase, of the decision making process.

For a *yes/no-issue* the contributions that can be made are not related to enlarge or to reduce the solution space, but to reveal one's opinion to the particular solution or option at hand. In a conversation people can have a positive, negative or neutral stance regarding statements or Y/N-issues. For this purpose the labels *'Positive'*, *'Negative'* and *'Uncertain'* are introduced. With the aim to reveal whether contributions from participants are either supportive, objective, or unclear. The *positive* relation can exist for example between a *yes/no issue* and a *statement* that is a positive response to the issue or between two *statements* agreeing with each other. When one speaker states that cows can be eliminated as being the most intelligent animals and the response from another participant is that cows don't look very intelligent, then the relation between these *statements* is *positive*. The *negative* relation is logically the opposite of the *positive* relation. It is to be applied in situations where speakers disagree with each other or when they provide a conflicting *statement* as a response to a previous *statement* or a *negative* response to a *yes/no issue*. In a case where it is not clear whether a contribution is positive or negative, but that there exists some doubt on the truth value of what the first speaker said, one should use the *uncertain* relation. From experience with the annotations it appears that in most cases it can easily be seen by the annotator whether the remark is mostly agreeing or mostly showing doubt.

A specification occurs in situations where a question is asked by one of the speakers and someone else asks a question which specializes the first question resulting in a possible solution space with more constraints. The contribution 'Which animal is the most intelligent?' can be specialized with the following proceeding contribution 'Is an

ant or a cow the most intelligent animal?' which again can be specialized if one for instance asks 'Are ants the most intelligent animal?'. For these occasions we introduce the label *'Specialization'*. The *specialization* label can for instance be applied when a particular issue generalizes or specializes another issue. It could on the other hand also very well happen that a person is not yet satisfied with the information or the argument explained. This person can explicitly invite the previous speaker to elaborate on his earlier *statements*. For these situations we define the relations *'Request'* in case someone asks for more information and the relation *'Elaboration'* if a person continues his previous line of thought and adds new information to it.

Whenever the issue is defined, an exchange of ideas about the possible answers or possible solution naturally occurs in the decision making process. Whenever a *statement* is made as a response to an *open-issue* or an *a/b-issue* it might reveal something about the position of a participant in the solution space. In general he provides an *'Option'* to settle the issue at hand. For example when a speaker asks 'Which animal is the most intelligent?' and the response from someone else is 'I think it's an ant' the *option* relation is to be applied. The opposite of the *option* relation is the *'Option-exclusion'* relation, and it is to be used whenever a contribution excludes a single option from the solution space.

The final relation of our set is to be applied when the content of a particular contribution is required to be able to figure out whether another contribution can be true or not. We named this the *Subject to* relation, which is somehow related to the concession relation in Toulmin's model. It is to be applied for example in the situation where someone states that 'If you leave something in the kitchen, you're less likely to find a cow' and the response is 'That depends if the cow is very hungry'. So the second contribution creates a prerequisite that has to be known before the first contribution can be evaluated. If the cow is very hungry the support could be either *positive* or *negative*. The *uncertain* label is not to be applied it this case, as the stance of the person in question is clear once the prerequisite is filled in. The *uncertain* label is merely to be used when an issue is preceded by a request or a specialization.

2.2.3 The Structure

TAS was constructed in a way that it preserves the conversational flow. TAS aims to keep as much chronology information of a discussion available in the argument diagram. By applying a left-to-right, depth first search, walk through on the resulting trees, the reader is able to read the resulting trees as if reading transcripts [Rienks and Verbree, 2006]. This is realized by assuring that in principle every contribution of a participant becomes a child of the previous contribution, unless the current contribution relates more strongly to an ancestor of the previous contribution. If a contribution is more strongly related to an ancestor of the previous contribution then it is to the previous contribution it self, the branch containing the previous contribution is 'closed' and contributions that follow can not be 'added' to it anymore.

Figure 2.2 depicts an example of a tree with closed branches. The numbers depicted in the nodes are based on their ranking on time, so in principle there would be relations between node 1 and 2, node 2 and 3, node 3 and 4 and node 4 and 5, unless one of the target nodes mentioned would relate more strongly to a node which is not its source node nor is on a 'closes' branch. Still node 4 is more strongly related to node 2 than to node 3 so we have created a relation between node 2 and 4 instead of node 3 and 4. This means at the same time that we have 'closed' the branch with only node 3 in it. It is not anymore allowed to relate nodes to node 3. In the same way we have created a

relation between node 1 and 5, thus closing the branch containing node 2, 3 and 4. If we would encounter a new node, node 6, we would only be allowed to relate it to node 5 or 1.In this way the reader still is able to read the tree as if reading a transcript.

An example of such a situation in which we deviate from the principle of relating an utterance to the previous one we find in figure 2.1. The utterance of P3: "And solar cells , I dunno about that ." is uttered right after the phrase "Yeah , I don't think it would" as well by P3. Still they are not related because the first utterance relates much stronger to the issue which started the discussion: "three different types of batteries . Um can either use a hand dynamo , or the kinetic type ones , you know that they use in watches , or else uh a solar powered one ." P3 makes it clear that he now focuses his opinion on solar cells instead of the "kinetic type" mentioned before.

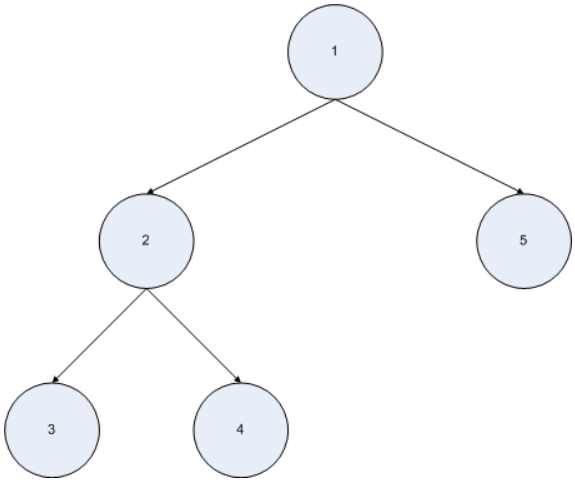

Figure 2.2: Example of a diagram with closed branches

To annotate the HUB corpus we have manually selected all discussions out of each meeting, thus creating a corpus of 241 discussions. Three annotators annotated this corpus of discussions resulting in a total of 8281 nodes and 4889 relations. The outline of the nodes in our corpus is shown in table 2.2, where the outline of the relations is shown in table2.4. Furthermore table 2.3 shows the differentiation of the relations over the source and target nodes.

Number of utterances	*Label*
4245	Statement
199	Weak Statement
244	Open Issue
72	A/B Issue
460	Yes/No Issue
3061	Unknown
8281	Total

Table 2.2: An overview of the nodes identified in the HUB Corpus

Source / Target	*Open Issue*	*A/B Issue*	*Yes/No Issue*	*Statement*	*Weak Statement*	*Total*
Open Issue	29	12	41	305	26	413
A/B Issue	2	0	10	86	6	104
Yes/No Issue	24	5	54	600	46	729
Statement	86	28	201	2347	77	2739
Weak Statement	4	2	12	180	19	217
Total	145	47	318	3518	174	4202

Table 2.3: An overview of the relations between different types of nodes identified in the HUB Corpus

Number of relations	*Label*
2028	Positive
408	Negative
218	Uncertain
490	Option
14	Option Exclusion
580	Elaboration
111	Specialization
183	Request
170	Subject To
4202	Total

Table 2.4: An overview of the relations identified in the HUB Corpus

2.3 Agreement

Most annotated corpora are (like the HUB corpus) annotated by several annotators. To measure the agreement amongst annotators mostly a few annotators the same part of the corpus. Measuring the agreement in annotations can be done because of three reasons: Measuring 1) reproducibility, 2) internal consistency *or* 3) learnability.

2.3.1 Reproducibility

Almost every annotation scheme is designed to be applicable to several corpora and to be used by different annotators. In these cases an annotation manual is developed to guarantee that annotators who have not been involved in the developing of the annotation scheme still would be able to apply the annotations correctly. Measuring agreement in cases as these gives insight in the interpretation of the annotation rules in practice. A high agreement signals that in when annotating other corpora the annotation will not highly depend on the choice of annotators.

In our work the reproducibility factor has been less important. The annotations have been performed by three annotators, who all had their influence in the developing of TAS as well. Annotations were made without first developing an annotation manual. Several individual annotations were reviewed and discussed by the group until their was a mutual understanding of the annotation scheme. Measuring internal consistency and learnability have been our main goals in measuring agreement.

2.3.2 Internal Consistency

The reproducibility of an annotation scheme might be a step beyond internal consistency. An internal consistent annotation of a corpus claims that an annotation is not influenced by an annotator, its mood or the amount of sleep he had the night before, but only by the data he/she has to annotate. A scheme is internal consistent if different annotators produce the same annotation of a discussion, regardless the condition they are in.

The internal consistency of the annotations have been important to us, because a high agreement would tell us that all annotations have been made 'the same', independent of the annotator. A low internal consistency could indicate a difference of opinion in the use of the annotation scheme and would in any case indicate that the lesser value of our annotations.

2.3.3 Learnability

A third concern involving agreement is the learnability of an annotation scheme. Annotating corpora is a time-expensive job and therefore computer annotators are highly welcome. Learnability is about the possibility of a system to *learn* to annotate using the specific scheme. As mentioned before a low agreement indicates a low internal consistency: annotators' interpretations of the rules described in the annotation scheme differ or other non-relevant circumstances (e.g. the personal situation of the annotator) have influenced the annotations resulting in inferior annotations. With such a possible low agreement between annotators or inferior annotations we can not expect a system, which is trained using the annotations made, to score a high performance on annotating. Measuring agreement can thus show what could be expected from a computer annotator at most, just as a baseline is used to define what performance at least could be expected.

Since our work concerns the learnability of classification of nodes in order to structure discussions the learnability of TAS is of great importance. This has been our biggest concern in measuring agreement.

2.3.4 Agreement in Segments, Nodes and Relations

The annotation of the HUB corpus have been done by three annotators. To measure the agreement amongst the annotators a subset of 12 discussions was chosen to be annotated by two annotators. The annotations of these discussions were used to give an estimate of the measure of agreement over all annotations.

There are several methods, with different goals for measuring agreement between annotators as the κ-measure by Cohen [Cohen, 1960]. This κ-measure can be used to measure the agreement when classifying distinct occurrences in distinct classes. The κ-measure and its results for TAS are described in section 2.3.5. Since our annotations concerned not only identifying the type of an utterance, but the utterance it self as well we have used a long side the κ-measure a different measure as well. This will be discussed in this section.

Our annotations concern *"the identification of a segment, labeling it with a type to construct a node and placing nodes in labeled relations to each other"*, thus a measurement of agreement should address all of these things as well. Therefore we have to distinguish three different terms: segments, nodes and relations.

- A segment is a part of the discussion which is labeled as a coherent set of words. So each discussion consists of a few to a lot segments.
- A node is a segment with a label (statement, weak statement, etc.).
- A relation is a 'line' which can be drawn between two nodes. Each relation has a label (positive, negative, etc.) as well.

The HUB corpus as we have annotated it consists of literal transcriptions of the discussions which are being represented chronologically. Because we are dealing with discussion selected from multiparty meetings this means that occasionally two or even more people speak at the same time. This means that (because of the chronological ordering of the transcription) one speaker turn (in the transcription) is being interrupted by someone else's speaker turn. This makes it hard to always stick to our terminology, therefore in this report report the words *segment*, *node* and *utterance* will all be used and will mean more or less the same, unless the context makes it clear that there is a difference in the meaning. Most of the times an *utterance* is the same as a *segment* both identifying the text available in a *speaker turn*. The main difference between a *segment* or *utterance* and a *node* is the label which is included in a *node* and not in the other two.

In figure 2.3 three different segments structures are displayed. The upper structure displays just one segment. The middle structure displays three different segments. The lower structure displays a segment which is being interrupted by another segment. An example of a transcription having such a structure can be found in table 2.5.

Figure 2.3: Three different segment structures

Because of the possible interruption of segments which can be at many places we have decided not to calculate the κ-measure over segments, but to calculate the agreement in *'segment starts'*, which is based on the NIST-SU Metric [Ang et al., 2005]. To achieve this we have listed all the starttimes of the segments and compared these two lists (for each annotator one). In figure 2.4 two annotations are visualized. The first annotator annotated three segments, where the second annotator annotated two. This results in the equal number of starttimes for each annotator. The agreement is then calculated by computing the percentage of agreed segments on all identified segments. In this example the agreement would be $\frac{1+1}{2+1} \times 100\% = 66.6\%$ In the same way the agreement in nodes and relations is computed. The tables 2.6, 2.7, 2.8 and 2.9 show these agreements in segments, nodes and relations. These tables show us that the agreement on segment boundaries is quite high, although annotators seem to differ in the length of an segment. For the structure of our discussion this does not have to be a

P1:	*there's there's a em emerging market for sort of touch screen L_C_D_ remotes that can be uh programmed in m much more sophisticated ways than sort of conventional models , so you get the sort of you get um you [other] you can redesign the interface to your own needs , you can programme in macros , and you get a much greater degree um um I mean you get in these sort of [other] three in one , five in one , whatevers , but you can get integration between the different uh the the the diff the different things that it's designed to control , to a much greater extent , and you can have one uh you know one macro to turn the uh you know turn the T_V_ to the right channel , get the uh re uh rewind the tape in the V_C_R_ and get it to play once it's rewound , for instance*
P0:	*Okay .*
P1:	*. Um b it occurs to me there might be a niche for uh for a remote that aimed towards some of that sort of functionality but using a just conventional push button design . And therefore putting it into a um well much lower price bracket .*

Table 2.5: Example of interrupted segment

Figure 2.4: Two segment annotations

problem. If we for instance would find the utterances *'So, that's my idea.'* and *'I truly think that the idea is pretty bad, just look at the colors. And apart from that, I believe we should come up with a better shape than a banana.'*, we might identify the first utterance as *statement*. The second utterance seems to be multi-interpretable, it might be classified as one *statement* having a *negative* relation to the first utterances, or as two *statements* both having a *negative* relation to the first utterance. In both cases TAS would be applied correctly, although the second approach would be preferable. But this example does show that the number of segments identified does not significantly have to influence the representation. Still one should note that it is very well possible to end up with several diagrams from one discussion as there are likely to be more than one possible interpretation. Walton [1996] for instance showed that various different argument diagrams can be instantiated by one single text. Moreover, in Rhetorical Structure Theory (RST) [Mann and Thompson, 1987], which is perhaps one of the theories closest to TAS, suggest that the analyst should make plausibility judgements rather than absolute analytical decisions, implicating more than one reasonable analysis.

Furthermore we can conclude that the agreement in the assigned types is over 70%, which seems to be a reasonable score. A further analysis of the difference of opinions on the resulting nodes is discussed in section 2.3.5.

Finally we have also researched the agreements on annotated relations. The figures in table 2.9 show us that there is a very low agreement in relations. Since the classification of relations is beyond our work of classifying the utterances we have not further researched this.

Annotator 1	531 segments
Annotator 2	622 segments
Agreement in starts of segments	71.47% (412 starts)
Agreement in start en ends of segments	56.46% (315 segments)

Table 2.6: Agreement in segments

Annotator 1	531 nodes
Annotator 2	622 nodes
Agreement	38.68% (223 nodes)

Table 2.7: Agreement in nodes (same segment, same label) - comparison 1

2.3.5 κ-measure

As mentioned before, the κ-measure can be used when one wants to measure the agreement in classifications of distinct occurrences in distinct classes. κ simply is *"the proportion of agreement corrected for chance"* [Cohen, 1960].

Cohen has not only introduced the κ-measure, but the weighted κ-measure as well. This weighted κ-measure can be used when the classes aren't ordinal. This means that class A *looks more alike* class B than it *looks alike* class C. Though this is this the case in our project (a *weak statement* is more like a *statement* then an *open issue*) we didn't compute this weighted κ-measure, since we are not in any way able to define or compute the distance between two classes. Still we are able to use the *unweighted* κ-measure over the classification of nodes. To do this we first generated a diffusion-matrix out of the discussions which were annotated by different annotators to measure the agreement. A diffusion-matrix as we use it shows the classification of the same segment by two different annotators. We do not speak of a confusion-matrix because we can not take one annotators work as a standard. In this diffusion-matrix depicted in table 2.10 all the segments that were recognized by both the annotators were used. This means that some piece of text was recognized as a segment, still the annotators could disagree on the label of the segment (which makes it a node).

This diffusion-matrix shows us that there is a large disagreement in classifying nodes which could be of the type *unknown* as well as the type *statement*. This mostly is caused by phrases which can be interpreted as backchannels or as an agreement, such as *'Yeah'* . Because of the fact that our annotations were done on the transcripts a lot of additional information, particularly phonetic cues as prosody are left out. The κ-measure for the annotations according to this diffusion-matrix is 0.4977, which is pretty bad. Therefore we have calculated the κ-measure for three other diffusion-matrices as well. Since the most disagreement was found in labeling phrases which could be interpreted as well as backchannels as agreements, and this had a great influence because of our small data set we decided to calculate alternate κ-values as well. This was not just done because it would benefit our results, but because of the fact that this disagreement shows us that our rules on how to annotate phrases as *'Yeah'* were not sufficient: New agreements need to be made on the classification of such ambiguous utterances. Therefore we could eliminate this problem in three ways:

1. Classify the label according to the judgement of Annotator 1: Let's say that Annotator 1 his opinion about this phrases is according to the agreements that

Annotator 1	531 segments
Annotator 2	622 segments
Matching segments	315
Matching nodes	70,79%

Table 2.8: Agreement in nodes (same segment, same label) - comparison 2 - (223 out of 315 identified segments with the same start- and endtime are labeled the same)

Annotator 1	203 relations
Annotator 2	313 relations
Agreement in start and end nodes	8,91% (23 relations)
Agreement in start and end nodes and label	2,52% (13 relations)
Agreement in label **if** agreement on start and end nodes	56.52%

Table 2.9: Agreement in relations

should have been made.

2. Classify the label according to the judgement of Annotator 2: Let's say that Annotator 2 his opinion about this phrases is according to the agreements that should have been made.

3. Delete all the segments which were labeled as Statement by Annotator 1 and as Unknown by Annotator 2: Because of the disagreement on how to label such phrases (not the actual labeling it self) it is better just to eliminate them from our matrix.

The three diffusion-matrices which are the outcome of these elimination-solutions are shown in tables 2.11, 2.12 and 2.13. The (unweighted) κ-measures which correspond to these matrices are 0.8789, 0.9054 and 0.8686. These κ-measures show that the agreement is above 0.87 which is a better result than the mentioned 0.50 before and even is a satisfactory indication that our corpus can be labeled with high reliability using TAS [Carletta, 1996]. This high agreement amongst human annotators encourages us in the research for a method of automatically classifying these utterances.

Although we were not able to calculate the relation between the different classes and thus weren't able to calculate the weighted κ-measure we can conclude that that the actual (weighted) κ-measure would even be better than the calculated κ-measure presented here. This because of the fact that a class as *weak statement* is closer to the class *statement* than it is to *open issue*.

To conclude this section about the HUB-corpus and its annotations is is good to mention that the calculated κ-measures are not really representative for the annotation of the corpus. Since we only have compared the annotations of 2 annotators for an amount of 12 discussions, which is about 5% of the corpus. Furthermore it is the case that in none of the compared discussions the label *a/b issue* was used. To overcome this problem we also have computed κ-measures over *virtual annotators*. This will be addressed in chapter 6, although it must be said that these *virtual annotators* can not be seen as some sort of replacement for the κ-measure.

Annotator 1 / Annotator 2	Statement	Weak statement	Open issue	A/B issue	Yes/No issue	Unknown
Statement	**75**	6	0	0	4	4
Weak statement	0	0	0	0	0	0
Open issue	0	0	4	0	2	0
A/B issue	0	0	0	0	0	0
Yes/No issue	0	0	0	0	0	0
Unknown	**70**	0	0	0	1	172

Table 2.10: Original diffusion-matrix

Annotator 1 / Annotator 2	Statement	Weak statement	Open issue	A/B issue	Yes/No issue	Unknown
Statement	75	6	0	0	4	4
Weak statement	0	0	0	0	0	0
Open issue	0	0	4	0	2	0
A/B issue	0	0	0	0	0	0
Yes/No issue	0	0	0	0	0	0
Unknown	0	0	0	0	1	242

Table 2.11: A diffusion-matrix in which annotator 1 'wins'

Annotator 1 / Annotator 2	Statement	Weak statement	Open issue	A/B issue	Yes/No issue	Unknown
Statement	145	6	0	0	4	4
Weak statement	0	0	0	0	0	0
Open issue	0	0	4	0	2	0
A/B issue	0	0	0	0	0	0
Yes/No issue	0	0	0	0	0	0
Unknown	0	0	0	0	1	172

Table 2.12: A diffusion-matrix in which annotator 2 'wins'

Annotator 1 / Annotator 2	Statement	Weak statement	Open issue	A/B issue	Yes/No issue	Unknown
Statement	75	6	0	0	4	4
Weak statement	0	0	0	0	0	0
Open issue	0	0	4	0	2	0
A/B issue	0	0	0	0	0	0
Yes/No issue	0	0	0	0	0	0
Unknown	0	0	0	0	1	172

Table 2.13: A diffusion-matrix in which ambiguous nodes are deleted

Chapter 3

Features

In the previous chapters we have made clear what the title of this report: *On the Structuring of Discussion Transcripts Based on Utterances Automatically Classified* actually means. We have presented a corpus of discussions as well as TAS, the annotation scheme we have applied to this corpus. Furthermore we have shown some figures about the agreement amongst annotators using TAS on the HUB corpus, which have given a satisfactory indication that our corpus can be labeled with sufficient reliability, if better agreements on the classification of words *'Yeah'* would be made. Having described our task, our corpus and the annotation scheme it is now time to explain how we would want to achieve this.

TAS, being an argument representation scheme for discussions is based on the underlying semantics of a discussion. For human annotators understanding the meaning of an utterance most of the times is a piece of cake, even understanding expressions as the one just used. Computers on the other hand are symbol-machines, they can do about anything with the syntax, but will never *understand* an utterance like humans do.

But although classifying an utterance according to TAS is a semantic-based task it does not mean it has nothing to do with syntax. If we for example look at the utterance *'Would you like some coffee?'* then we are dealing with a question. We do not only know this is a question because on some meta-level we can conclude this, but also because of the words *'Would you'*. We have learned that an utterance which starts with a verb followed by personal pronoun usually is a question. So the syntax (*'Would you*) cues us for the semantics. We could say that the semantics of an utterance are somehow expressed in the syntax of the utterance and therefore the syntax can work as a cue for the semantics. This preassumption is the basis for feature extraction. Still we need to understand that classification of utterances is a context related task. One of the best examples might be in the classification of utterances as *weak statements*. It might be that in some cultures it is thought of as having good manners when one uses words as 'probably' or 'perhaps' in expressing one's opinion. In such cases it would not be good to use these words as cue words for a *weak statement*, since in this case these words do not cue for a specific type, but are present because of the cultural context. Thus it is not the case that each utterance will or should always be classified the same.

In feature extraction we are concerned with finding syntactic cues for classification. An ideal feature would only be available for utterances of type X, this would be an optimal-distinguishing feature. Unluckily for us, such features are sparse. We have to look for syntactic cues which distinguish our utterances as good as possible. These

cues could be words as in the example above, but a pitch rate or the number of words in an utterance as well. For all utterances we can extract the same features and these feature sets combined with the labels assigned to the utterances are then used to train a classifier. In chapter 5 we will write more about classifiers, for now the most important is that we are using syntactic information to get to the semantics of an utterance.

To extract our features from the TAS annotated HUB-corpus we made use of several Perl scripts which were able to handle the HUB XML-format and gather the right statistics from it.

In the remainder of this chapter we will describe the features extracted from our corpus, namely *Sentence Length*, *? and OR*, *Last Label*, *Ngram Points* and *POS Ngram Points*.

3.1 Sentence Length

Even a shallow study of our corpus, such as in table 3.1 shows us that most utterances which are labeled with the type *unknown* have very few words. This specific characteristic of utterances of the type *unknown* (which mostly are backchannels) makes it attractive to have a *sentence length* feature. The *sentence length* feature is defined by the number of tokens in the utterance and a low score on this feature could be a nice cue for utterances of the type *unknown*.

Tokens are constructed in a tokenization process. In this process a text is split up into tokens. A token is defined as an accepted character string. In most cases this will be a word. Tokenizers can have several additional options and features like a list of non-tokens: tokens which according to the given pattern would be recognized, but by declaring them as non-tokens automatically are discerned.

The tokenization used in this project is quite simple, by declaring each word as a token, where whitespace is used to define the border between words. Punctuation as ., *?* and *!* are defined as tokens as well.

Label	**Average**	**Standard deviation**
Statement	21.58	26.58
Weak statement	26.29	24.90
Open issue	23.07	24.68
A/B issue	34.06	35.43
Yes/No issue	22.17	19.30
Unknown	3.51	6.81

Table 3.1: Utterance length statistics (based on number of tokens)

3.2 ? and 'OR'

In common language most issues are (presented as) questions. To indicate that people are posing a question they can use their intonation in speech and question marks in written texts. This presence or absence of a question mark could therefore be a good indicator for an issue, as also can be concluded from table 3.2.

TAS makes a distinction between three sorts of issues, namely *Open issues*, *A/B issues* and *Y/N issues*. Furthermore *A/B issues* are defined as explicitly making it clear

that there is a **limited** set of options. A common way of presenting such a limited set of options is by using the word *or*. We have therefore also chosen for a feature which indicates the presence of the word *or* in an utterance. Table 3.2 also shows that utterances of the type *a/b issue* most frequently have as well an '?' as an 'or' present.

Label	**?**	**OR**	**? and OR**
Statement	3.20	25.30	1.56
Weak statement	2.51	31.16	1.01
Open issue	84.42	28.28	22.95
A/B issue	77.78	70.83	55.56
Yes/No issue	81.52	32.83	24.13
Unknown	1.27	2.38	0.20

Table 3.2: Percentage of utterances containing the token(s) mentioned

3.3 Last Label

In conversations and discussion people do not only try to create syntactically and semantically correct utterances, but also try to let their utterances be of an interesting contribution to the conversation. This means that people do not just utter things occasionally, but tend to react to one another. This gives us reason to state that the intention or function assigned to an utterance is related to intentions and functions assigned to previous utterances in a conversation. Since a 'Dialogue Acts'-like scheme like TAS tries to model the intention which underlies utterances, we can *translate* this assumption in 'the label assigned to an utterances is related to labels earlier assigned'. Table 3.3 shows the coherence of labels assigned to two chronologically successive segments. The same information is graphically shown in figure 3.1. It shows the chance of segment X being of type A given that the previous segment, segment Y, was of type B. We can for example conclude that based on these figures the chance of a segment being of type *Unknown*, given that the previous segment was of type *Yes/No issue* is 0.2473.
These figures show that our given the label of a node the chance of the (chronologically) next node of being of some type is not equal to the unconditional chance of a node being of some type. A few interesting things to see are for example that the chance of *weak statement* being followed by *weak statement* is bigger then 'unconditionally' would be expected. Perhaps do *weak statements* 'trigger' each other. This is not the case for *a/b issues* followed by other *a/b issues* or *open issues*. Not one *a/b*

First node/Second node	**Statement**	**Weak statement**	**Open issue**	**A/B issue**	**Yes/No issue**	**Unknown**
Statement	0.5785	0.0212	0.0224	0.0089	0.0534	0.3154
Weak statement	0.5888	**0.0761**	0.0203	0.0051	0.0660	0.2437
Open issue	0.5926	0.0206	0.0412	0.0041	0.0947	0.2469
A/B issue	0.5556	0.0412	**0.0000**	**0.0000**	0.0556	0.3472
Yes/No issue	0.6110	0.0503	0.0306	0.0088	0.0525	0.2473
Unknown	0.4194	0.0216	0.0302	0.0080	0.0495	0.4712
None (start)	**0.2241**	0.0083	**0.1411**	**0.0249**	**0.1249**	0.4772
Unconditional	0.5126	0.0240	0.0295	0.0087	0.0556	0.3696

Table 3.3: Chronological transformations of nodes

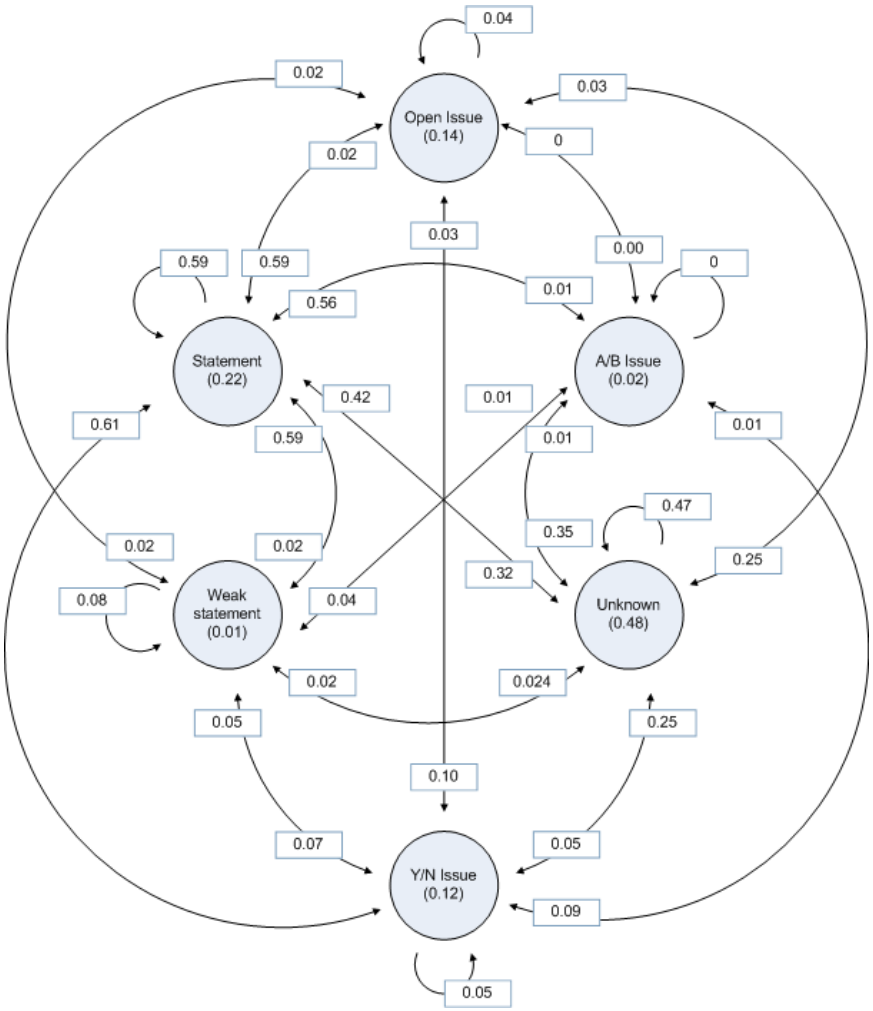

Figure 3.1: Chance of transformations of nodes in time. The chance of a start node being of specific type is in parenthesis

issue is followed by another *a/b issue* or *open issue*. Finally we see that discussions are more often started with issues than could be expected from the unconditional chance of an issue appearing, which of course could be expected because something (mostly an issue) needs to be discussed.

We see that the type of a node seems to be related to the type of the previous node. Our assumption still is that not only an utterance in influenced by the previous one, but by its full 'history'. We have performed experiments of using 1 to 4 previous labels as a basis for classification. Using the two previous labels of utterances mentioned right before the current segment our classifiers performed best. Therefore we introduce the *Last label* feature as a set of two features describing the two previous labels.

Another reason for having the *Last label* feature is the 'ambiguity' of text in the transcripts of the HUB corpus. In the corpus 2127 nodes are non-unique, meaning that

exactly the same text of an utterance is differently classified. The only feature which in such cases could make a distinction is the *Last Label* feature. If we would not have this feature we could only predict the majority class of utterances with the specified text, which would result in 619 incorrect classified utterances. Table 3.4 shows 8 of the most ambiguous phrases.

Let us for example look at the utterance *"yeah ."*. This specific utterance occurs 479 times labeled as *statement*, 354 times labeled as *unknown* and once labeled as a *weak statement*. Without the *Last Label* classifier we would predict the majority class for this utterance (*statement*) and would therefore classify this phrase 355 times incorrect.

utterance	*statement*	*weak statement*	*unknown*
yeah .	479	1	354
okay .	119		83
right .	44		27
mm-hmm .	21	3	322
mm .	14	3	256
yeah , yeah .	17		10
yes .	11		36
yep .	10		18

Table 3.4: A few of the most ambiguous utterances

3.4 Ngram Points

Just as question marks have a high chance of indicating questions and issues other keywords and keyphrases might very well indicate types as well. For example it is very likely that words as 'probably' and 'perhaps' have a high chance of indicating the type *weak statement*. In dialogue act classification much research has been done finding words and phrases which can work as an indicator. Hirschberg and Litman [1993] for example present a long list of cue phrases with words as 'further' and 'therefore'. These cue phrases are manually selected after having studied numbers of conversations and are presented as context-free cue phrases.

Another way of finding cue phrases is by automatically selecting them. Reithinger and Klesen [1997] are one of the first who use this approach in dialogue act classification. Their approach is based on ngrams. An ngram is a subsequence of a sentence consisting of n consecutive tokens. First they created all ngrams for each utterance in the corpus. Then they used the number of occurrences of each ngram in the corpus. The more ngram X occurs in utterances of type A and the less it occurs in utterances of other types then type A, the bigger the chance that X is a cue phrase for A.

Using the Ngram Statistic Package (NSP) by Banerjee and Pedersen [2003] we have constructed all bi- (subsequences of two consecutive tokens), tri- (subsequences of three consecutive tokens) and quadrigrams (subsequences of four consecutive) for each utterance. This package has the possibility of constructing ngrams using several options as a stoplist, non-tokens etc. For each type we count the number of ngram-occurrences. An example of the counts of three trigrams are shown in table 3.5. If shows that the trigram 'what do you' occurs 3 times in an utterance of the type *state-*

ment, 14 times in an utterance of the type *Open issue*, 4 times in an utterance of the type *A/B issue* and 2 times in an utterance of the type *Yes/No issue*.

trigram	statement	weak statement	open issue	ab issue	yn issue	unknown
what do you	3	0	14	4	2	0
do you think	3	1	8	4	24	0
we have to	47	3	3	1	8	4

Table 3.5: Examples of trigram occurrences

Having counted each ngram we want to select only the most predictive ones. Several experiments have been set up to develop a good method of 'predictive ngram selecting', they will be discussed in chapter 4. For now we will not concern how we have attributed points to each ngram for each type but will accept these as given. The actual value for the ngram feature is a vector with size 6. For each type we sum all the points for that type for each ngram in the utterance. This we do for uni,- bi-, tri- and quadrigrams.

Let us illustrate this by an example. Suppose we are to classify the utterance *"And what else do you do with the jog-dial ?"* (which is of the type *Open issue*. First we create all uni-, bi-, tri- and quadrigrams which gives us the ngrams listed in the first column of table 3.7. Then for each ngram found we look up the points it contributes in the ngrams-table based on our training set. These figures can be found in the other columns in the table. To conclude we compute the total score for each ngram-order and type, which we do by just adding up each individual score. So the actual value of this feature is not just one value, but actually is a combination of 4 times (each order) 6 value (each type), as presented in table 3.6. As we can see the type *statement* scores the best, but this does not directly mean that it will be classified as a *statement* since our classifier will use all features of all training data to classify a new instance.

Using this approach of automatically acquiring cue phrases and attributing points to them we are more likely to find context-specific cue phrases which might work as a good indicator for the semantics.

3.5 POS Ngram Points

The use of cue phrases seems to be a good feature for classifying dialogue acts according to studies as performed by Samuel [2000] and therefore we have used it as one of our features as described. But one of the problems of cue phrases is that they might be too specific and therefore miss the power of generalization. A possible solution for this problem is to make use of the part-of-speech of a word.

Each word belongs to a *part-of-speech*, like noun, pronoun or adverb. In school most children learn what tokens belong to what part-of-speech and having some experience part-of-speech tagging seems fairly easy. Still even very simple words as 'book', might be ambiguous and therefore harder to tag than might be expected. 'book' should be tagged as a noun in an utterance like "I've read this book", but should be tagged as a verb in an utterance like "Are you going to book that flight?"

The last 30 years several methods have been developed to perform automatically part-of-speech tagging and have made part-of-speech tagging from a point of interest in NLP to a research area in it self. Performing part-of-speech tagging always starts with constructing a new or selecting an existing tag set. A tag set is a set of tags

ngram order	statement	weak statement	open issue	ab issue	yn issue	unknown
unigram	6815	295	876	238	1331	406
bigram	128	4	41	11	59	24
trigram	2	0	0	0	0	0
quadrigram	0	0	0	0	0	0

Table 3.6: The ngram feature values

belonging to lexical classes. Most tag sets are based on the tag set used for part-of-speech tagging the Brown Corpus [Francis, 1980]. The Brown Corpus was the first major corpus of English for computer analysis and consists of 500 samples of randomly chosen publications. One of the tag sets based on this tag set is the Penn Treebank tag set [Marcus et al., 1993]. It consists of 45 tags, where the Brown corpus tag set consisted of 87. Several syntactic distinctions are not marked in the Penn Treebank tag set which made it possible to use fewer tags. In our work we tagged our utterance with the Stanford Log-Linear POS tagger which uses the Penn Treebank tag set.

There are generally three different approaches for automatic POS-tagging: rule-based, stochastic and transformation-based. The Stanford Log-linear part-of-speech tagger is a maximum entropy tagger which is a stochastic tagging variant [Toutanova et al., 2003]. The Stanford Log-linear POS tagger is a state of the art tagger developed in Java with an accuracy of 97.24% on the Penn Treebank WSJ. This good result together with its good performance on unknown words make it a good POS-tagger for our project.

Returning to our first concern: why would we want to use POS-information? To make use of the power of generalization. We could for example have found cue ngrams like in the list below.

- What do (as in *What do you like?*)
- Which would (as in *Which would you choose?*)
- Why can (as in *Why can't I come with you?*)

This leaves us with three different cue phrases which all cue for the type *Open issue* or *A/B issue*. But by making use of part-of-speech tags we find a more general applicable cue phrase, namely WP VBP (a Wh-pronoun followed by a Verb, non-3rd person singular present).

Just as with our feature *Ngram Points* we have constructed all POS-ngrams for each utterance and type, and attributed them with points to construct a value for our *POS Ngram Points* feature. As far as we know using ngrams part-of-speech tags has only be done in research on creating backchannels in spoken dialogues [Cathcart et al., 2003].

ngram	statement	weak statement	open issue	a/b issue	y/n issue	unknown
and	1278	46	61	28	108	64
what	241	7	149	14	57	30
else	15	2	9	3	6	6
do	189	8	62	20	85	19
you	1754	63	116	28	222	91
do	189	8	62	20	85	19
with	354	16	23	9	49	17
the	2667	140	180	62	335	128
jog	1	0	0	0	0	0
dial	2	0	0	0	1	0
?	125	5	214	54	383	32
TOTAL	6815	295	876	238	1331	406
and what	7	0	1	1	2	0
what else	1	0	3	1	0	5
else do	0	0	0	0	0	0
do you	19	0	32	9	41	4
you do	8	0	0	0	2	5
do with	8	0	0	0	0	0
with the	84	4	5	0	14	10
the jog	0	0	0	0	0	0
jog dial	1	0	0	0	0	0
dial ?	0	0	0	0	0	0
TOTAL	128	4	41	11	59	24
and what else	0	0	0	0	0	0
what else do	0	0	0	0	0	0
else do you	0	0	0	0	0	0
do you do	0	0	0	0	0	0
you do with	0	0	0	0	0	0
do with the	2	0	0	0	0	0
with the jog	0	0	0	0	0	0
the jog dial	0	0	0	0	0	0
jog dial ?	0	0	0	0	0	0
TOTAL	2	0	0	0	0	0
and what else do	0	0	0	0	0	0
what else do you	0	0	0	0	0	0
else do you do	0	0	0	0	0	0
do you do with	0	0	0	0	0	0
you do with the	0	0	0	0	0	0
do with the jog	0	0	0	0	0	0
with the jog dial	0	0	0	0	0	0
the jog dial ?	0	0	0	0	0	0
TOTAL	0	0	0	0	0	0

Table 3.7: An example of getting the value for the ngram features

Chapter 4

Classification

In the previous chapters we have described TAS, the HUB Corpus and the features we can extract from each segment. This chapter is about a subtask of applying TAS to a discussion: the classification of utterances. Classifying utterances as TAS nodes is a task very similar to Dialog Act Tagging, therefore we have applied the method presented in this report in Dialog Act Tagging as well. This will be described in chapter 8.

Although human annotators apply TAS by simultaneously classifying nodes as well as relations, our approach is of splitting up these tasks into subtasks, to decrease complexity. The advantage of classifying the way human annotators do is that at any stage the context of a node or relation to classify (i.e: the nodes and relations near) is available and can be used as information for classifying. For our automatic classification of nodes we have made a split up of the classification task: first we will classify all the nodes and then the relations. But still to make sure that we do not miss too much information of the context we have introduced the *Last label* feature. With this feature we keep track of the last labels assigned to previous utterances.

To classify our nodes we have computed all the features for each node in our corpus. Several experiments have been done on our created *feature sets* using different settings and training and test sets. The remaining of this chapter will concentrate on the setup of the experiments performed.

4.1 Balanced and Unbalanced

In the discussion on the HUB corpus table 2.2 showed us that our corpus is quite unbalanced. Nodes of the type *statement* and *unknown* together populate 88% of the corpus. This means that obtaining a good results on our corpus would be possible if a classifier only would be able to find a good distinction between *statements* and *unknowns*. Perfectly indicating nodes of type *A/B issue* in this case is far less important than classifying *statements*.

In our work we do not only aim for good results on the total corpus but are interested in the results for the different types as well. Therefore we have chosen not only to perform experiments on our original unbalanced corpus, but also on a balanced version. This we have done in two different set ups: a *unbalanced-on-balanced* setup and a *balance-on-balanced* setup. Our first setup consisted of an unbalanced training set and a balanced test set. We constructed 10 test sets containing 50 randomly picked nodes

of each type, resulting in a test set of 600 nodes. The remainder of the corpus was used as the training set, thus having constructed an unbalanced training set and a balanced test set.

For our *balanced-on-balanced* setup we first randomly picked 70 lines of each type out of our corpus, constructing a corpus of 420 lines. Using this subcorpus we constructed a 10-fold train/test split, constraining the test set (and thus the training set as well) to have an equal number of nodes for each type.

Using these two different setups we hope to show the influence of the *'balanced-ness'* of the training set on the classification of the test set.

4.2 Ngram Selection Methods

In the previous chapter on feature extraction table 3.7 presented us an example of how to compute the scores for each ngram-order and type. These score should be analyzed by our classifier to identify the relations between these scores and the actual label of an utterance. Classifying our nodes on basis of these points means that the ngrams identified in our training set and the points attributed to them are crucial to our performance. By counting all the ngrams per type in the training set we create what in speech recognition traditionally is called a 'language model' (LM) per type. Table 4.1 gives the scores for the different ngram-order/type combinations based on the example in table 3.7, in which all the ngrams in the training set were used to score points. In this same example we saw that ngrams as 'jog' or 'dial' though having few occurrences were considered as cue phrases. Furthermore our training set could incorporate ngrams which might have many occurrences but can hardly be found cue phrases as 'a' or 'this'. Since we enumerate points of all cue ngrams, we can not leave the selection up to the classifier. Our classifier will not be able to determine the influence of each specific cue ngram. Therefore we believe it would be good to perform a selection on the ngrams found in the training set by deleting the 'non-cue ngrams' and so hopefully boosting our performance. This assumption is partly based on the work of Webb et al. [2005] which is to our knowledge the only research using *ngram selection methods*. In the remainder of this section we will present several of these ngram selection methods.

Order/Type	statement	weak statement	open issue	ab issue	yn issue	unknown
1	6815	295	876	238	1331	406
2	128	4	41	11	59	24
3	2	0	0	0	0	0
4	0	0	0	0	0	0

Table 4.1: Example of points obtained for each ngram-order and type

4.2.1 Normalizing Ngram-Values

Our first ngrams selection method, normalizing ngram-values, actually is not an ngram-selection method, but it does influence the predictivity of an ngram. It strikes us that in general the occurrences of an ngram found in utterances of the type *statement* are much bigger than the others. Two basic explanations are that 1) there is a very high amount of statements in our corpus and therefore a specific ngram will have more chance to be

encountered in a node of type *statement* and 2) typically *statements* have more words than for example nodes of type *unknown*, which as well increases the chance that a specific ngram will be encountered in a node of type *statement*.

To overcome this potential problem one variation on our ngram-points table might be to normalize it over all ngrams for each type. Would we for example have counted 1000 ngrams of the type *statement* and 100 ngrams of the type *weak statement* then the *unigram-statement*-score for the ngram 'and' would be 1.28 whereas the *unigram-weak statement*-score would be 0.46 (see table 3.7). By normalizing our ngram-values we hope to overcome the problem that utterances of the type *statement* can score more points on ngram-values for each type, just because of the fact that they in general have more words and therefore a bigger chance of having more cue ngrams, regardless of what type they cue for.

4.2.2 Select1/3Normalized

Commons sense tells us that not every ngram in an utterance will be a cue for the type of the utterance. Words as 'and' do not particularly act as a cue as words as 'which' or 'perhaps' do. To include non-relevant ngrams will therefore only create noise so non-relevant ngrams should be eliminated from the list of ngram-point-values. By selecting only the ngrams which claim to have a certain predictivity we hope to improvre our classification performances.

An ngram which is a cue for a specific type should relatively have significantly more occurrences in nodes of this type than in those of other types. Or select1/3Normalized (s1/3n) method is based on this preassumption. S1/3n selects only an ngram if

- it occurs at least 3 times in the training set
- after normalizing the occurrences per type for the number of ngrams per type (just as calculated in section 4.2) it has one or more types with a normalized occurrence-ratio which is equal or bigger than $\frac{1}{3}$ of the total normalized occurrences.

Example: Table 4.2 shows three ngrams found in a specific corpus. The total of ngrams found for each type is given. To check which ngrams will be selected we first calculate the total number of occurrences for each ngram. The ngram 'this' occurs in total 255 times, 'the' 405 times and 'perhaps' 46 times, so each ngram meets our first constraint.

For our second constraint we first have to normalize our counts for the total of ngrams found for each type, these are shown in table 4.3. We can see that 'the' occurs significantly more in nodes of type *yes/no issue*, since $\frac{0.197}{0.284} > \frac{1}{3}$ and thus the ngram 'the' will be selected. The same goes for the ngram 'perhaps', which appears $\frac{0.006}{0.0151} \approx 0.397$ of the time in nodes of the type *weak statement*. Unlike the ngrams 'the' and 'perhaps', the ngram 'this' will not be selected, because even for the class in which it appears the most, *yes/no issue* it does not meet our $> \frac{1}{3}$ treshold.

Order/Type	statement	weak statement	open issue	a/b issue	y/n issue	unknown	total
this	100	60	20	3	45	15	243
the	200	100	20	11	59	5	395
perhaps	10	30	5	1	0	0	46
Total ngrams	10,000	5000	1000	250	3000	2000	21,250

Table 4.2: Example of ngrams found in a corpus

Ngram	statement	weak statement	open issue	ab issue	yn issue	unknown	total
this	0.01	0.012	0.02	0.012	0.015	0.008	0.077
the	0.02	0.02	0.02	0.044	0.197	0.003	0.284
perhaps	0.0001	0.006	0.005	0.004	0	0	0.0151

Table 4.3: Example of ngrams found in a corpus, normalized for the total number of ngrams found per type

4.2.3 DROPn

Unlike the Select1/3Normalized method our DROPn method does not require an ngram to 'occupy' a certain amount of the total ngram-space, but is requires an ngram to completely 'ignore' nodes of a certain class. By selecting ngrams which fulfill such a requirement we will have ngrams with more distinctional power, thus having the chance of improving our classification performance.

An ngram will be selected if

- it occurs at least 3 times in the training set
- there are n types for which the ngram occurrences is 0.

Example: using the DROP1 method only the ngram 'perhaps' (table 4.2 will be selected, just as when using the DROP2 method.

We have performed experiments with the DROP1 and DROP2 ngram selection methods. More on this can be found in chapter 6.

4.2.4 TOPx

A third ngram selection method first ranks all ngrams and then selects the top X. By selecting only the most predictive ngrams a lot of 'noisy' ngrams (ngrams which only cue a little for a certain type, but cue for another type as well) might be eliminated, which hopefully improves our classification performance.

To rank the ngrams for each ngram a ranking score is computed. This computation of this score is based on the following assumptions:

- Of two ngrams, the ngram which is more biased (after normalization) to a specific type (i.e. it relatively occurs more in the type in which it occurs the most then the other ngram occurs in the type it occurs the most) has a bigger probability of being a cue phrase than the other.
- Of two ngrams that are 'equally biased' (after normalization) to a specific type the one which occurs more often in nodes of the specific type has a bigger probability of being a cue phrase than the other.

The ranking-score described is equal to the product of the number of times the ngram occurs in nodes of type X *and* the part of this 'ngram-space' occupied by nodes of type X.

Example: The score for the ngram 'this' (see table 4.2) for type *statement* is $\frac{100}{100+60+20+3+45+15} \times 100 = 41.15$, for the ngram 'the' the score is $\frac{200}{200+100+20+11+59+5} \times 200 = 101.27$, and the ngram 'perhaps' is valued $\frac{10}{10+30+5+1+0+0} \times 10 = 2.17$. So our ranking of ngrams for the type *statement* would be 'this', 'the', 'perhaps'.

We have performed experiments with this ngram selection method selecting the top 10, 25 and 50 ngrams for each type. More on this can be found in chapter 6.

4.3 Compressed or Individual

In chapter 3 we described our way of getting points for our ngram features, optionally by using a *ngram selection method* as described in this chapter. Using the ngrams available (which can be all ngrams in the training set or the ngrams selected by one of the ngram selection methods described above) points were obtained for each type and ngram order. By doing this we were able to drastically decrease our number of features. Would we, for example select the 10 best ngrams for each ngram order and not sum up all the points obtained per type, then alone for ngrams of words we would need a maximum 10 ngram features per type per order which is a total of $10 \cdot 6 \cdot 4 = 240$ features. Of course it could be possible that some ngram would be in the top-X list of ngram for more than one type, but still we would need a lot of features, where our setup requires only $6 \cdot 4 = 24$ features. So using our method we are able to reduce the feature set with a factor equal to the number of ngrams selected.

To investigate the performance of our method of enumerating the points over all ngrams selected we have also performed experiments using the same ngrams, each as an *individual feature*, with the number of its occurrences in an utterance as its value.

4.4 The Order of Ngrams

Until now we have treated each ngram order as it in generally would carry as much information as the other ngram orders. Hence our TOPx ngram selection method was applied to ngrams all of the same order, thus selection x unigrams, x bigrams, x trigrams and x quadrigrams. Still it could be and it is not very unlikely that our $x + 1$th unigram (the first one that has not been selected) scored better than one or more of our top x quadrigrams. To investigate this we have also performed experiments using the top x ngrams regardless of their order, using $x = 10$, $x = 25$ and $x = 50$.

Chapter 5

Classifiers

In the previous chapters we have described the features we have extracted from our training and test data to perform classification. Furthermore several conditions for experiments have been presented in order to get to the best performance of classifying utterances. The result of these processes have been a great list of values for each utterance together with the type it belongs to. In this chapter we will describe our way of classifying the utterances on basis of the features extracted: The task of the classifier is to find the relation between the values extracted and the label an utterance belongs to.

For our classification task we made use of Weka. Weka is a collection of machine learning algorithms for data mining tasks [Witten and Frank, 1999]. Learning the correct labels for all the nodes on the basis of *feature values* (called *attribute values* in Weka) is a specific data mining task for which WeKa is suitable. It is designed to perform clustering and classification and is used in many NLP-tasks. Several classifiers can be chosen and each classifier has its own advantages. We have used several classifiers in our experiments, namely J48, DecisionTable and Multilayer Perceptron. For each classifier we will give an overview on how it works. For more in depth description, see [Quinlan, 1993, Kohavi, 1995, Minsky and Papert, 1969, Werbos, 1974].

5.1 J48

J48 is a decision tree classifier. Decision trees represent a supervised approach to classification. A decision tree consists of non-terminal nodes and terminal nodes. The non-terminal nodes represent tests on one or more attributes of the data. The terminal nodes represent the outcome: the decision of the classifier. At each non-terminal node the classifier will test an attribute of the input instance and push it on a branch depending on the outcome of the test. Finally a terminal node will be reached, which is the decision of the classifier.

A lot of work on decision trees has been done by [Quinlan, 1993], who has developed several decision tree models. C4.5 is an implementation of one of Quinlan's models. This model is also available in the Weka classifier package and is named J48. The general approach performed by J48 is as follows[1]:

1. Choose an attribute that best differentiates the output attribute values.

[1]Taken from `http://grb.mnsu.edu/grbts/doc/manual/J48_Decision_Trees.html`

2. Create a separate tree branch for each value of the chosen attribute.
3. Divide the instances into subgroups so as to reflect the attribute values of the chosen node.
4. For each subgroup, terminate the attribute selection process if:
    a All members of a subgroup have the same value for the output attribute, terminate the attribute selection process for the current path and label the branch on the current path with the specified value.
    b The subgroup contains a single node or no further distinguishing attributes can be determined. As in (a), label the branch with the output value seen by the majority of remaining instances.
5. For each subgroup created in (3) that has not been labeled as terminal, repeat the above process.

Figure 5.1 depicts a part of a tree constructed by J48 based on two features: *length* and *? and or token*. It shows pretty clearly that at each node the data is either split up and pushed on specific branch (ellipse) or assigned a label (rectangle). In this way all data is processed and assigned a label. Since J48 at every step splits up the dataset on basis of one attribute normalizing the data does not influence it's performance: all the values for the specific attribute will be normalized and therefore the mathematical relations will still be the same.

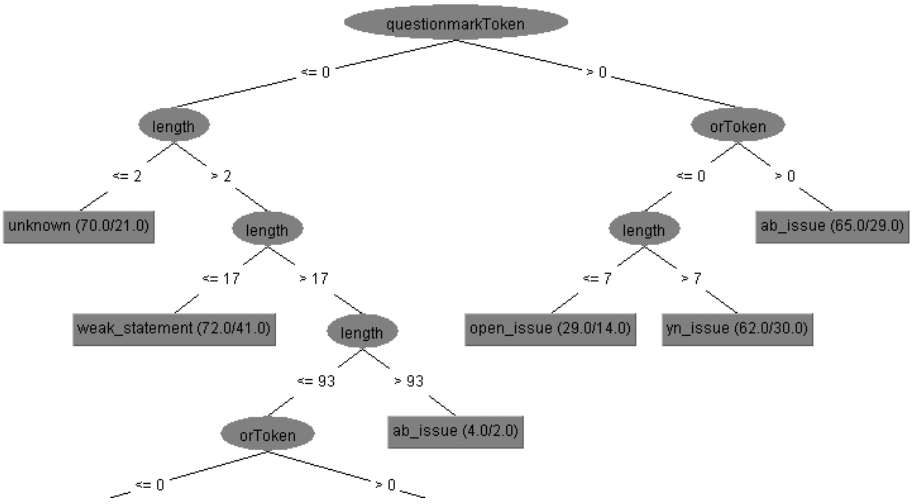

Figure 5.1: An example of a part of a J48 tree

5.2 DecisionTable

A decision table is a rule-based classifier. It checks for the validity of several conditions and takes actions if conditions are fullfilled. A standard decisiontable looks like presented in table 5.1.

	Condition 1	Y	Y	Y	Y	N	N	N	N
Conditions	Condition 2	Y	Y	N	N	Y	Y	N	N
	Condition 3	Y	N	Y	N	Y	N	Y	N
	Action 1	X			X				
	Action 2		X	X	X				X
Actions	Action 3				X		X		
	Action 4					X	X		
	Action 5							X	

Table 5.1: A standard decision table

One of the advantages of decision tables for humans is that it's structure makes it almost impossible to forget a special case. Because of the boolean character of the decision table every combination of conditions can be checked and linked with a set of actions. Kohavi [1995] describes the Inducer of Decision Table Majority (IDTM) which searches for the subset of features with the minimal error when used as condition attributes in a Decision Table.

This IDTM algorithm is used in Weka as well. Table 5.2 depicts a Decision Table as given by Weka using the same train/test set as described in section 5.1. In this case the feature subsets with the minimal error turned out to be the feature set in total. All instances which can not be classified using the conditions in the DecisionTable will be classified as instances of the majority class (in this case *statement*).

Non matches covered by Majority class.

questionMarkToken	orToken	length	label
$(-\infty - 0.5]$	$(0.5 - \infty)$	$(2.5 - \infty)$	weak_statement
$(0.5 - \infty)$	$(0.5 - \infty)$	$(2.5 - \infty)$	ab_issue
$(-\infty - 0.5]$	$(-\infty - 0.5]$	$(2.5 - \infty)$	weak_statement
$(0.5 - \infty)$	$(-\infty - 0.5]$	$(2.5 - \infty)$	yn_issue
$(-\infty - 0.5]$	$(-\infty - 0.5]$	$(1.5 - 2.5]$	unknown
$(0.5 - \infty)$	$(-\infty - 0.5]$	$(1.5 - 2.5]$	open_issue
$(0.5 - \infty)$	$(0.5 - \infty)$	$(-\infty - 1.5]$	open_issue
$(-\infty - 0.5]$	$(-\infty - 0.5]$	$(-\infty - 1.5]$	unknown
$(0.5 - \infty)$	$(-\infty - 0.5]$	$(-\infty - 1.5]$	yn_issue

Table 5.2: A Decision table for QMT_ORT_L by WeKa

Let us for example classify the first utterance in 2.1: "three different types of batteries . Um can either use a hand dynamo , or the kinetic type ones , you know that they use in watches , or else uh a solar powered one ." In this example we find 0 times the token '?', 2 times the token 'or' and we find the utterance existing of 36 tokens. Unluckily for us this utterance will be (according to the first line) classified as a *weak statement*, where it is annotated as an *a/b issue*. It would be labeled that way by our classifier if the utterance would have contained a '?'.

5.3 MultilayerPerceptron

Neural networks are functional classifiers. They are based on the parallel architecture of the animal brain and are a form of a multiprocessor computer system. A Multilayerperceptron (or feedforward neural network) is a special kind of neural network, consisting of multiple layers of perceptrons.

A perceptron is simple binary classifier which maps its inputs x_i or x (a real-valued vector) to an output value $f(x)$. With x as the input-vector, w as the weighting-vector (where w_i the weight 'attached' to x_i) and b as the bias. The output $f(x)$ of a perceptron is

$$\sum_{i=1}^{j}(w_i \cdot x_i) + b$$

The classification of x is based on the sign of the output.

In many applications the perceptrons apply a sigmoid function as an activation function. This sigmoid function makes it possible for a neural network to compute a continuous output instead of a step function. A sigmoid function produces a curve having an "S" shape. Several sigmoid functions do exists, such as the arc-tangent, the hyperbolic tangent and the logistic function. The logistic function is used very often and is defined as

$$\varphi(x) = \frac{1}{1 + e^{-}x}$$

A schema of a perceptron is depicted in figure 5.2.

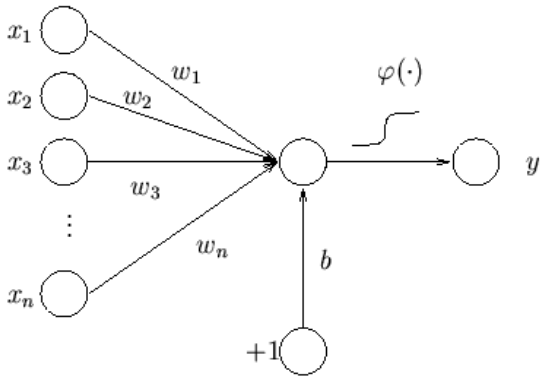

Figure 5.2: A schematic picture of a perceptron

In 1969 Minsky and Papert [1969] showed that it was impossible for these classes of network to learn an XOR function. They conjectured that similar results would hold for multilayer perceptrons . They proved out to be wrong. In the 1980's it was discovered that multilayer perceptrons were able to learn an XOR function and many other functions. A schematic overview of such a (feedforward) multilayer perceptron is shown in 5.3. In this figure there is an input layer (the green label), a hidden layer (the red nodes) and an output layer (the yellow nodes).

A feedforward multilayer perceptron can be learned by the backpropagation algorithm. The roots of backpropagation lay in the phd-thesis of Werbos [1974]. Over the

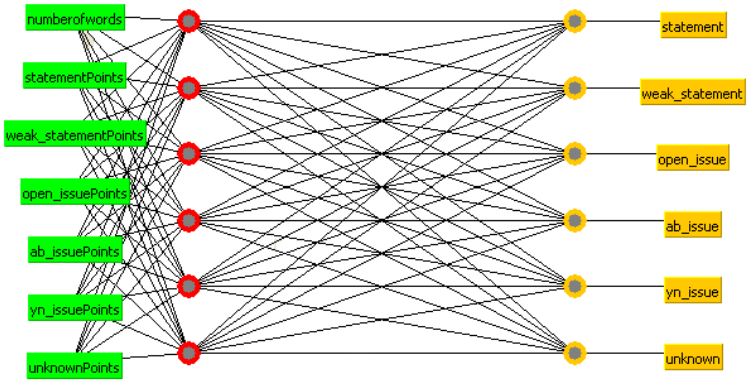

Figure 5.3: An example of a neural network

years the algorithm has become one of the most popular learning algorithms for neural networks. The summary of the technique is as follows[2]:

1. Present a training sample to the neural network.
2. Compare the network's output to the desired output from that sample. Calculate the error in each output neuron.
3. For each neuron, calculate what the output should have been, and a scaling factor, how much lower or higher the output must be adjusted to match the desired output. This is the local error.
4. Adjust the weights of each neuron to lower the local error.
5. Assign "blame" for the local error to neurons at the previous level, giving greater responsibility to neurons connected by stronger weights.
6. Repeat the steps above on the neurons at the previous level, using each one's "blame" as its error.

To adjust the weights properly a general method for non-linear optimization task that is called gradient descent is used. For this, the derivation of the error function with respect to the network weights is calculated and the weights are then changed such that the error decreases (thus going downhill on the surface of the error function). For this reason back-propagation can only be applied on networks with differentiable activation functions, which explains the necessity of the sigmoid function.

A possible problem with multilayer perceptrons is that the network can overfit the training data. In such a case the network does not generalize anymore, it does not structure the true statistical process, but is structures all training instances, including the noise. Another problem is that sometimes a neural network can get 'stuck' in a local minimum of the error function, instead of searching further for the global minimum. Luckily there are a lot of simple practical solutions to these little problems.

[2] http://en.wikipedia.org/wiki/Backpropagation

5.4 Choosing the Right Classifier

Choosing the right classifier for a task like ours is very hard to do. Each classifier has its pros and cons and it is almost impossible to make an analysis of the classifier best fit to classify the data at hand. Therefore we have started out with a few basic experiments on different classifier to select a decent one. The classifier involved in these experiments were BayesNet, Ibk, Kstar, Logistic, MultilayerPerceptron, SMO, J48 and DecisionTable. MultilayerPerceptron, J48 and DecisionTable all came with good equivalent results and were therefore selected as potential good classifiers.

Still, some small conclusions can be made in the comparison of the classifiers. Kohavi [1995] claims that *"IDTM outperforms C4.5 in discrete domains where the features interact and not too many features are relevant. Decision trees are well suited for local relevances (i.e., different features are relevant in different regions of the instance space), but the greedy top-down recursive partitioning algorithms tend to fail when features interact. DTMs are suited to concepts where some features are globally relevant; the feature subset selection algorithm used here is conducting a best-first search and is thus able to capture interactions."*

The *interaction* of features should be understood as features related to and influencing each other. Such a (slight) interaction might have existed between the features *or* and *length* if we would have chosen the *or*-value to be the number of times the token 'or' occurred in an utterance. This because longer utterances in general might have more 'or' tokens. Still, we have the *or*-value chosen to be 0 or 1, according to the absence or presence of an 'or' token, thus eliminating possible interaction between the features.

This absence of interaction, combined with the fact that J48 is far more faster than the other two classifiers made us decide to initially run all experiments using the J48 classifier. Experiments with promising results where then run again using the other classifiers. Still the emphasis on our work is in the selection of the right features and not of the right classifiers.

Chapter 6

Results

Up to now we have shown TAS, the HUB corpus, the features we use to classify the utterances, several experiments and the classifiers used, but no one result has been presented yet. In this chapter we will first discuss the baseline we used and then present several results obtained in different experiments.

6.1 Baseline

To measure whether or not our classification method is good, plain results do not suffice, but they have to be compared to earlier results. In our case there are no earlier results. Until now TAS have only been applied to the HUB corpus and our work is the first on automatically applying TAS to discussions. To be able to 'compare' our results we here use a baseline, which is a very simple classification method.

One of the most simplest (and most frequently used) baselines is to predict the largest class. This can be done without any feature extraction on the training set. In our work we have automatically selected our cue phrases, using several ngram selection methods. Therefore *another* interesting baseline would be one based on manually selected cue phrases. Samuel [2000] introduces such a baseline, called the LIT set, in his work on DA Tagging using Transformation Based Learning. This set consists of 687 different cue phrases presented in twelve papers, dissertations and books [Cohen, 1987, Fraser, 1990, Grosz and Sidner, 1986, Halliday and Hasan, 1976, Heeman et al., 1998, Hirschberg and Litman, 1993, Knott, 1996, Marcu, 1997, Reichman, 1985, Schiffrin, 1987, Warner, 1985, Zukerman and Pearl, 1986].

In chapter 2 we showed that the HUB corpus is quite unbalanced (see table 2.2). Because of this unbalancedness good performances are very easy to obtain, for instance by classifying everything as part of the majority class. Therefore we presented the idea of a balanced corpus in chapter 4. By also computing results for a balanced corpus we hope to show that although our method may perform less on those train/test set combinations they significantly outperform the baseline. Different baseline results have been computed for the two different setups of a balanced corpus. Table 6.1 shows the different baseline performances.

The remainder of this chapter will present and discuss the results obtained in several experiments using different settings. We will start by discussing the unbalanced corpus, followed by a section on a balanced test set using a classifier trained on an unbalanced training set and conclude with a section on a balanced test set using a classifier trained

Corpus (train/test) / Baseline	Majority class	LIT
Unbalanced / Unbalanced	51.26	71.79
Unbalanced / Balanced	16.67	37.10
Balanced / Balanced	16.67	45.71

Table 6.1: Different baseline performances for the TAS utterance classification of the HUB corpus

on a balanced training set as well. All results using the J48 classifier can be found in the table in this chapter. These tables present no values for the different ngram orders and the feature set using the non-order specific ngram ranking. This was alone because using the non-order specific ngram ranking it would be 'unfair' to judge it for a specific ngram order. The results for classifiers using only a subset of the features *Question Mark Token*, *Or Token*, *Length* and *Last Label* are only presented once since they are not influenced by the ngram selection method used.

6.2 Unbalanced-Unbalanced

As discussed in chapter 4 we have performed several experiments using different ngram selection methods. Furthermore experiments using different feature sets have been performed. All results of these experiments using an unbalanced training as well as test set are shown in table 6.2. Our best performance using the J48 classifier is 78.52%, but more interesting conclusions can be drawn from these results. All further analysis of the results in this chapter concentrates on the results obtained using the TOPx ngram selection method.

Aantal instances	LM	NORM	S1/3N	DROP1	DROP2	Sum-T10	Sum-T25	Sum-T50	Indiv-T10	Indiv-T25	Indiv-T50	sumAll10	sumAll20	sumAll50	indiAll10	IndiAll25	indiAll50
L	68.92																
QMT_ORT_L	71.83																
QMT_ORT_L_LL	74.10																
P1	71.08	71.08	71.08	51.26	51.26	71.11	70.87	71.08	71.51	71.54	71.63						
P2	69.92	69.92	69.92	65.11	69.92	70.79	70.32	70.14	70.95	71.46	71.22						
P3	67.96	69.96	67.96	68.04	67.03	60.95	63.59	65.15	60.86	63.30	64.09						
P4	68.33	68.33	68.33	66.98	67.06	51.44	54.99	57.58	51.26	53.75	55.26						
W1	71.44	71.44	71.44	64.16	71.44	70.22	71.32	70.57	72.68	73.13	73.07						
W2	69.79	69.79	69.79	68.74	69.79	56.66	60.28	63.77	56.04	59.59	62.30						
W3	67.62	67.62	67.62	64.23	67.62	51.77	51.83	51.77	51.43	51.41	51.44						
W4	57.17	57.17	57.17	48.61	57.17	51.33	51.25	51.39	51.26	51.26	51.26						
P[1234]	71.05	71.05	71.05	69.68	69.10	70.60	69.93	69.09	71.73	71.73	71.03	71.37	71.16	67.73	71.48	71.54	71.73
W[1234]	67.70	67.70	67.70	68.34	68.66	69.98	70.73	70.78	72.85	72.90	72.77	70.44	70.86	71.18	72.69	73.41	72.90
P[1234]W[1234]	68.59	69.02	68.57	70.67	70.92	73.31	73.60	73.36	75.93	75.53	75.39	74.28	73.77	73.23	76.04	76.02	75.75
L_P[1234]W[1234]	68.89	68.89	68.89	70.75	70.82	73.92	73.32	73.87	76.44	75.99	75.58	74.70	74.47	73.76	76.19	76.02	74.71
QMT_ORT_L_P[1234]W[1234]	70.64	70.64	70.64	72.76	73.18	74.98	74.60	74.74	76.54	75.96	75.56	74.59	74.94	75.02	76.01	76.13	75.03
QMT_ORT_L_LL_P[1234]W[1234]	72.33	72.33	72.33	74.64	74.98	77.20	76.98	76.38	78.52	78.13	77.89	76.97	77.03	76.86	77.83	78.49	77.27
LIT	71.79																

Table 6.2: Comparison of results on an unbalanced-unbalanced set using different features and parameters

6.2.1 Ngrams of Words vs. Ngrams of POS-tags

First of all we see that in general uni- and bigrams of words (the 4th 5th line of table 6.2 perform better in classification that uni- and bigrams of POS-tags (line 8 and 9), but tri- and quadrigrams of POS-tags (line 10 and 11) seem to perform better than those of words (line 6 and 7). This probably is because tagset consists of 36 tags resulting in for example $36^4 = 1679616$ possible quadrigrams, which is quite much, but not a much as our unlimited set of possible quadrigrams of words. The chance of more than once encountering 4 specific words in a row is far less than encountering 4 specific POS-tags in a row: quadrigrams of POS-tags are for more general and are therefore expected to work better as a feature than quadrigrams of words. On the other hand this generalization of POS-tags negatively influences performance using uni- and bigrams. Each uni- and bigram of POS-tags occurs so much in any type, that they hardly cue for a specific type, where uni- and bigrams of words do.

6.2.2 Combinations of Ngrams of Words and POS-tags

If we look at the classifiers using all ngrams of POS-tags (line 12, P[1234]) we notice that the choice of ngram selection method hardly influences the performance, which is not the case for classifiers based on all ngrams words. Selecting the right ngrams of words certainly seems to influence our results and given the performances on classifiers using unigrams of words it is highly likely that the right selection of unigrams of words contributes the most to a good performance.

Furthermore it is interesting to see that when using classifiers do not use any ngram selection method (LM, NORM) some interference seems to occur between all the ngrams of POS-tags and all the ngrams of words, which explains the loss of performance compared to the individual performances of the combined features. The same seems to occur when using a classifier based on our Select1/3Normalized method. All other classifiers seem to benefit from the combination of as well all word ngram features as POS ngram features. From this we can conclude that selecting the most predictive ngrams does not only improve performance, but even is necessary if a classifier wants to benefit from using both features concerning ngrams of words **and** POS-grams, instead of just one of the two.

6.2.3 Ngram Based Features vs. Non-Ngram Based Features

Three features that we use are not based on ngrams, these are the features *Length (L)*, *Last label (LL)* and *Question Mark Token / Or Token (QMT_ORT)*. Interesting enough is that using these features alone already gives us a good performance (68.92, 71.83, 74.10). This is because (just as we expected when describing our features) *especially* the *Length* feature seems to be a good classifier for *statements* and *unknowns* which have a high presence in the HUB corpus. Again we see that only the classifiers using ngram selection methods (except from the Select1/3Normalized method) benefit from the combination of *Ngram based features* and *Non-ngram based features*. The performances when using as well *Ngram based features* as *Non-ngram based features* ($L\_P[1234]W[1234]$, $QMT\_ORT\_L\_P[1234]W[1234]$, $QMT\_ORT\_L\_LL\_P[1234]W[1234]$) are in any case better than the individual performances (L, QMT_ORT_L and QMT_ORT_LL compared to $P[1234]W[1234]$). The classifiers not using any ngram selection method (as well as S1/3N) do not benefit from the combination of these features, for these classifiers the features seem to

interfere with each other.

6.2.4 Summed vs. Individual

To judge the use of a 'compressed' feature set as we did we have compared the performance results to results based on the same ngrams, but without the enumeration of points over types. Our results show that the use of individual ngrams as features instead of the enumeration of points over these ngrams almost always increases the performance (varying from 1 to 3 percent). This improvement of performance seems to be uninfluenced by the number of ngrams selected. This was to be expected because making use of the ngrams as individual features gives us a lot more information on an utterance than when enumerating the occurrences of an ngram over each type in the training set.

6.2.5 Order-specific of Ngrams vs. Non-Order Specific Ngrams

All experiments using selected ngrams regardless of their order have had less ngrams selected than experiments using x ngrams for each type. If we compare the different results we see that using order-specific ngrams gives us slightly better results. This on the one hand could be because of the selection of more ngrams, but on the other hand we see the performance of classifiers using non-order specific ngrams decreasing over the number of ngrams selected. Therefore a better explanation is that the higher order ngrams which are available in the order-specific ngrams classification gives this little bit of extra classification power. In most cases we are overfitting our training data by constructing tri- and quadrigrams that should act as cue phrases. Most tri- and quadrigrams normally will not be available much in the training set as well in the test set and so normally will score less in the ranking than uni- and bigrams. But the few which are available in as well the training as test set seem to be not that specific (and 'overfitted') as the others and may improve the performance a bit. These tri- and quadrigrams will not be selected using a non-order specific ngrams classifier, but will be using an order-specific classifier, thus explaining the slightly better performance of the latter.

6.2.6 Number of Ngrams to Select

In the previous section we indicated that increasing the number of ngrams for a classifier using non-order specific ngrams decreases performance. This is not unique for this classifier, but we see it with almost all classifiers: increasing the number of ngrams selected (slightly) decreases performance. Determining which is the best number of ngrams to select can not be done yet since we have not exhaustively tested our methods using different numbers of ngrams. But our results do show that the optimal number of ngrams to select seems to be around 10 for order-specific classifiers or some value between 10 and 25 for non-order specific classifiers. Figure 6.1 shows the results of the few experiments we have done on varying the number of ngrams to select. The increasing performance of the 'Non-Order Specific Individual' classifier using 25 ngrams is probably due to the fact that having selected to few ngrams (10) there are to few features for good classification and having selected to many ngrams (50) overfitting (modeling the noise) takes place. The 'Individual' classifiers are more sensitive to overfitting, because they can not 'compensate' it as the 'Compressed' classifiers can, by enumerating points over the ngrams available.

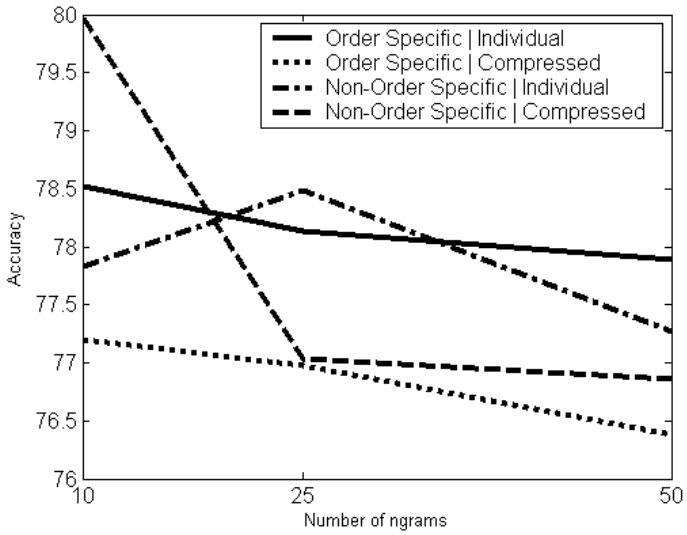

Figure 6.1: Classifier performance related to the number of ngrams selected for the QMT_ORT_L_LL setting

6.2.7 Most Interesting Results

Our best performance is 78.52%, but this is not the only interesting result. We elaborate on 6 selected interesting feature set/parameter combinations. Table 6.3 explains the abbreviations used in table 6.4 containing the feature set/parameter combinations we have selected as well as their performance.

Abbreviation	**Feature**
L	Length
P	Uni-, bi-, tri- and quadrigrams of POS-tags
W	Uni-, bi-, tri- and quadrigrams of words
QMT	Question mark token
ORT	'OR' token
LL	Last Label

Table 6.3: Features and their abbreviations

These feature sets/parameter combinations have been selected because of their good performance. For each feature set the best compressed and individual version were chosen, resulting in the above sets. For these sets we have computed their performance using the DecisionTable and MultilayerPerceptron classifiers as well. Comparing the results acquired by different classifiers we see that using the MultilayerPerceptron classifier on the feature set/parameter combinations in which we used the individual ngrams performances are far worse, where all other results are quite alike regardless of what classifier is used. Looking at the confusion matrices produced by these classifiers we see that no utterance is classified *open issue* or *yes/no issue* and that only a few are classified as *unknown*. A possible explanation for this might be

Feature set/parameter combination	J48	DT	MLP
L_P_W (Non-Order Specific) (Compressed Top 10)	74.70	73.80	74.36
L_P_W (Order Specific) (Individual Top 10)	76.44	75.70	38.41
QMT_ORT_L_P_W (Order Specific) (Compressed Top 10)	74.98	75.30	73.95
QMT_ORT_L_P_W (Order Specific) (Individual Top 10)	76.54	75.70	23.75
QMT_ORT_L_LL_P_W (Order Specific) (Compressed Top 10)	77.20	76.23	74.85
QMT_ORT_L_LL_P_W (Order Specific) (Individual Top 10)	78.52	76.71	39.56

Table 6.4: Classification performances for interesting feature set/parameter combinations for the unbalanced-unbalanced train/test set combination. (DT = DecisionTable, MLP = MultilayerPerceptron)

that MultilayerPerceptrons has the ability of 'combining' different feature values. The classifier might therefore be 'looking for' combinations of individual ngrams which cue for a label. Searching for such combinations might mislead a classifier, because of the fact that in general *statements* and *weak statements* have more words than other utterances and therefore more combinations, thus resulting in a biased classifier. Still we can not find an explanation for the fact that no utterance is classified as *open issue* or *yes/no issue*.

The performance of the feature set/parameter combinations mentioned in table 6.4 have also been computed for several other corpora on a Dialog Act tagging task, which is mentioned in chapter 9.

6.3 Unbalanced Training - Balanced Testing

In the previous sections we have described the most interesting characteristics of the results obtained in several experiment using as well an unbalanced training as test set. All performance of the unbalanced-balanced corpus are presented in 6.5. Here we will describe some interesting aspects of using a balanced instead of an unbalanced test set. The performances for the 'interesting' feature set/parameter combinations mentioned in table 6.4 synoptic presented in table 6.6 together with other interesting results for this particular train/test set combination.

Aantal instances	LM	NORM	1/3	DROP1	DROP2	Sum-T10	Sum-T25	Sum-T50	Indiv-T10	Indiv-T25	Indiv-T50	sumAll10	sumAll25	sumAll50	indiAll10	IndiAll25	indiAll50
L	26.03																
QMT_ORT_L	37.80																
P1	28.13	28.13	26.87	16.67	16.67	27.73	28.40	28.13	29.83	30.50	30.93						
P2	29.87	29.87	30.70	30.90	28.57	27.63	29.53	29.43	29.50	30.20	31.00						
P3	30.60	30.60	30.23	31.97	32.17	26.87	28.10	29.20	26.07	29.67	27.47						
P4	30.63	30.63	30.03	30.53	31.33	18.20	25.00	26.73	16.70	22.70	23.47						
W1	38.23	38.23	39.50	28.73	27.13	38.37	38.57	38.40	41.20	41.33	42.18						
W2	35.43	35.43	34.93	36.30	35.90	26.00	31.37	33.27	22.60	26.87	28.03						
W3	33.13	33.13	30.53	31.17	30.80	19.37	20.70	20.97	17.07	16.97	16.96						
W4	26.80	26.80	21.40	21.40	21.47	17.17	17.10	17.33	16.67	16.67	16.67						
P[1234]	31.27	31.27	31.47	31.73	33.17	31.20	30.80	31.80	32.77	33.07	33.13	28.23	30.50	30.53	29.33	31.67	34.13
W[1234]	32.97	32.97	37.90	35.17	35.20	38.10	38.27	38.20	42.07	42.40	42.33	38.97	39.50	39.17	42.13	41.53	41.80
P[1234]W[1234]	34.70	34.70	38.90	34.93	34.63	38.37	37.30	38.37	43.43	43.87	43.63	39.87	38.63	38.90	44.10	45.03	44.40
L_P[1234]W[1234]	34.57	34.57	38.77	34.60	35.00	39.00	37.47	38.73	43.40	44.15	44.23	39.97	39.30	38.70	44.17	45.17	44.50
QMT_ORT_L_P[1234]W[1234]	35.07	35.07	39.53	40.17	40.20	39.87	39.87	40.13	43.33	44.20	44.07	40.07	40.87	41.43	44.77	44.73	44.73
LIT	37.10																

Table 6.5: Comparison of results on an unbalanced-balanced set using different features and parameters

Feature set/parameter combination	Performance
L_P_W (Non-Order Specific) (Compressed Top 10)	39.97
L_P_W (Order Specific) (Individual Top 10)	43.40
QMT_ORT_L_P_W (Order Specific) (Compressed Top 10)	39.87
QMT_ORT_L_P_W (Order Specific) (Individual Top 10)	43.33
L	26.03
L_P_W (Non-Order Specific) (Individual Top 10)	44.17
L_P_W (Non-Order Specific) (Individual Top 25)	45.17

Table 6.6: Classification performances for interesting feature set/parameter combinations for the unbalanced-balanced train/test set combination.

6.3.1 The *Length* Feature

Using a balanced test corpus the *Length* feature seems to lose some of its significance. Because there are an equal number of utterances for each class in the test set a feature as *Length* which was of good use for making a distinction between the two largest classes now, still is but the overall performance is worse, because making a distinction between the type *statement* and *unknown* is not any more a guaranteer for good performance. Still we see that using the *Length* feature 26.03% of the utterances are classified correctly, which is significantly better than the majority class baseline of 16.67%.

6.3.2 Number of Ngrams to Select

Just as we saw with the unbalanced-unbalanced combination the number of ngrams selected slightly influences the performance. Our results show that the optimal number of ngrams to select seems to be between 25 and 50 for classifiers, regardless of the individual/compressed or order specific/non-order specific settings, although the number of ngrams selected hardly influences performance as can be seen in figure 6.2. The reason for this may be due to our balanced test set. Having created a balanced test set some classes have very few utterances in the training set to learn from. Increasing the number of selected ngrams might give the classifier just the extra information on classes with few instances to perform better.

6.4 Balanced Training - Balanced Testing

Up to now we have treated the unbalanced-unbalanced and the unbalanced-balanced train/test set combination. Here we will describe some interesting results obtained on a balanced-balanced train/test set combination. Performances for the 'interesting' feature set/parameter combinations mentioned in table 6.4 have also been calculated for this train/test set combination and are presented in table 6.7 together with other interesting results for this particular train/test set combination.

Although these performances may inspire us we must note that the size of the training set is very limited and therefore we can hardly speak of a 'language model', since it is only a combination of a 'few' ngrams. Still these results outperform the unbalanced-balanced train/test set combination and we would like to know why.

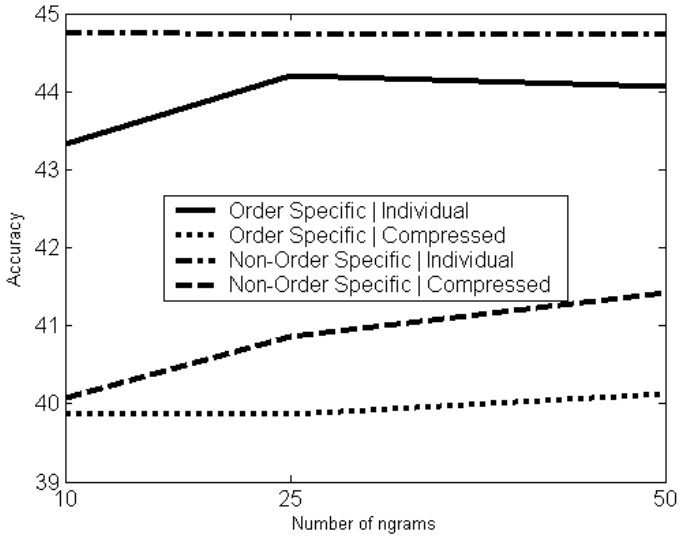

Figure 6.2: Classifier performance related to the number of ngrams selected for the QMT_ORT_L setting

6.4.1 Balanced vs. Unbalanced Training

Our results in table 6.8 show that in general using our balanced training set outperforms the use of the unbalanced training set. Comparing the different confusion matrices for the L_P_W (Non-Order Specific) (Individual Top 25) experiment (table 6.9 and 6.10) we see that this mostly is due to the more correct classification of classes with few instances, such as *a/b issue* and *weak statement* (see table 6.11). This is probably because using a balanced training set each class gets a 'honest' chance of being learned. Interesting to see as well is the improvement in performance using the features *Question Mark Token*, *Or Token* and *Length*. This is explained by the increased number of utterances of the types *open issue*, *a/b issue* and *yes/no issue* in the training set.

Feature set / parameter combination	**Performance**
L_P_W (Non-Order Specific) (Compressed Top 10)	37.62
L_P_W (Order Specific) (Individual Top 10)	47.62
QMT_ORT_L_P_W (Order Specific) (Compressed Top 10)	44.29
QMT_ORT_L_P_W (Order Specific) (Individual Top 10)	46.43
L_P_W (Non-Order Specific) (Individual Top 10)	48.33
L_P_W (Non-Order Specific) (Individual Top 25)	49.05
P_W (Non-Order Specific) (Individual Top 50)	51.43

Table 6.7: Classification performances for interesting feature set/parameter combinations for the balanced-balanced train/test set combination.

Number of instances	LM	NORM	1/3	DROP1	DROP2	Sum-T10	Sum-T25	Sum-T50	Indiv-T10	Indiv-T25	Indiv-T50	sumAll10	sumAll25	sumAll50	indiAll10	IndiAll25	indiAll50
L	28.33																
QMT_ORT_L	46.67																
P1	31.43	31.43	34.05	29.76	18.57	30.24	31.43	31.43	35.24	39.05	39.05						
P2	34.52	34.52	34.52	37.14	40.48	31.19	33.33	31.43	35.00	40.00	41.67						
P3	35.48	35.47	35.24	35.00	34.29	28.81	29.52	32.62	27.86	30.71	29.05						
P4	35.71	35.71	29.29	27.86	28.33	23.57	25.48	27.62	20.95	21.19	21.43						
W1	40.71	40.71	41.19	39.05	33.81	39.29	37.62	39.05	44.76	47.86	50.71						
W2	41.43	41.43	37.86	36.67	33.33	30.00	33.57	31.19	26.43	27.62	27.38						
W3	29.29	29.29	23.28	21.19	23.10	18.33	22.38	22.86	16.43	16.67	16.43						
W4	23.33	23.33	17.62	17.62	17.62	16.67	17.86	18.10	16.67	16.67	16.67						
P[1234]	33.81	33.81	34.52	36.43	40.95	30.00	35.24	32.62	38.81	42.38	39.76	31.43	33.57	34.29	36.19	38.81	42.43
W[1234]	34.76	34.76	40.24	38.33	38.33	41.19	40.24	41.67	46.67	48.10	46.19	42.67	40.95	39.05	45.24	48.10	49.05
P[1234]W[1234]	35.71	35.71	39.76	39.76	40.00	37.62	35.95	36.43	49.52	47.62	47.14	36.67	41.43	37.62	50.95	48.81	51.43
L_P[1234]W[1234]	36.19	36.19	39.52	44.76	36.43	39.52	36.19	37.62	**47.62**	46.19	46.67	**37.62**	39.76	37.86	48.33	47.86	50.95
QMT_ORT_L_P[1234]W[1234]	38.81	38.81	42.86	44.76	40.71	**44.29**	46.90	42.62	**46.43**	45.95	46.67	42.62	45.95	43.57	48.33	**49.05**	50.71
LIT	45.71																

Table 6.8: Comparison of results on an balanced-balanced set using different features and parameters

a	b	c	d	e	f	$<--$ classified as
4	**25**	**32**	**4**	**0**	**35**	**a = ab_issue**
2	55	17	7	0	19	b = open_issue
0	1	85	10	1	3	c = statement
0	0	29	71	0	0	d = unknown
0	**0**	**83**	**12**	**4**	**1**	**e = weak_statement**
2	11	26	8	1	52	f = yn_issue

Table 6.9: Confusion matrix of unbalanced-unbalanced train/test set

a	b	c	d	e	f	$<--$ classified as
41	**14**	**4**	**0**	**3**	**8**	**a = ab_issue**
11	37	5	2	2	13	b = open_issue
4	4	24	9	28	1	c = statement
1	2	14	43	6	4	d = unknown
5	**4**	**17**	**4**	**33**	**7**	**e = weak_statement**
13	19	4	0	6	28	f = yn_issue

Table 6.10: Confusion matrix of balanced-unbalanced train/test set

Train/test split	ab_issue	open_issue	statement	unknown	weak_statement	yn_issue
Unbalanced/balanced	4	55	85	71	4	52
Balanced/balanced	58.6	52.9	34.3	61.4	47.1	40

Table 6.11: Percentage of utterances per type correctly classified

6.5 Virtual κ-measure

In chapter 2 κ-measures were computed for the TAS annotation of the HUB corpus. Two problems met there were the small amount of discussions that could be compared and the absence of utterances of type *a/b issue* in each annotation. To get a bit more insight in the reliability of our corpus we performed experiments where the J48 classifier was trained using parts of the corpus annotated by one annotator (row) and was tested on a part of the corpus annotated by another annotator (column). This resulted in the performances shown in table 6.12. When both training and test sets were picked from the same annotator, we used 10-fold cross-validation.

These figures show that classifiers based on a model of annotator 1 or 2 perform quite reasonably, compared to the annotations of the other. Performances of models of annotator 3 perform worse, just as models of annotators 1 and 2 compares to the annotations of annotator 3 do. Interesting to see is that the model of annotator 1 performs the best on it self. We could say that annotator 1 is the easiest to model, which means that annotator 1 is most consequent in labelling utterances on basis of the feature values extracted. If on basis of these features an annotator should be able to classify an utterance then more agreement on annotating needs to be sought for annotator 3 and the other two. Still we must keep in mind that we are talking about annotations on basis of

Trained / Tested on	Annotator 1	Annotator 2	Annotator 3
Annotator 1	84.4%	75.7%	70.3%
Annotator 2	75.6%	79.5%	66.2%
Annotator 3	67.0%	66.2%	82.2%

Table 6.12: Performance amongst annotators

the features extracted. It might very well be possible that more or other features need to be extracted to model an annotator more correctly.

The main reason of measuring agreement concerns the learnability of TAS. The κ-value measured then was around 0.87, which indicated the best we could expect when making an comparison between our system and an annotator. To focus on the usefulness of our classifier is, we have computed the κ-values between the predicted outcome and the actual annotated data. As κ measures the degree of concurrence between two annotators the value could be described as *The degree of concurrence between annotator X and a model of annotator Y*. In this case the model of annotator Y is the model computed out of the training set and annotator X is the test set. The κ-values as computed by making use of the "QMT_ORT_L_LL_P_W (Order Specific) (Individual Top 10)-classifier" are shown in Table 6.13. First of all we notice that there are two κ-values for the inter-annotator agreement of two different annotators. This is because measuring the degree of concurrence between annotator X and a model of annotator Y is something else than degree of concurrence between annotator Y and a model of annotator X.

Furthermore it appears that the inter-virtual-annotator agreement is quite low. Although the maximum κ we could have expected is not 1, but 0.87 our κ's measured here still are quite low. This could be because of a lack of actual inter-annotator agreement. As described in chapter 2 annotations made were not based on an annotation manual, but on elaborate discussions on argument diagrams as constructed by the annotators. When they felt that there was an overall agreement on how to annotate using TAS, the HUB corpus was annotated. It might be that still, after all discussion, some annotators had misinterpretations on the annotating procedure to follow, resulting in *'inter-annotator disagreement'* and therefore the low κ-measures as presented. Another possible reason for our low κ-values might be due to the fact that our method of creating 'virtual' annotators perhaps is not sufficient for measuring concurrence between annotators, as mentioned above. And since our best classification result of 78.52% it self still leaves us with lot of incorrect classified instances we believe the last to be more likely than the first.

κ	Annotator 1	Annotator 2	Annotator 3
Annotator 1	0.68	0.57	0.50
Annotator 2	0.62	0.62	0.42
Annotator 3	0.46	0.44	0.69

Table 6.13: κ values for different annotator/virtual annotator combinations

Chapter 7

Identification and Classification of Relations

Although our work has mostly been on classifying nodes in order to present a meaningful discussion structure, a TAS-structure is described not only by nodes, but by relations as well. In chapter 4 we told that applying TAS is a two-step procedure: classifying nodes *and* identifying and classification of relations. In this chapter we will give an introduction on step 2: the identification and classification of relations. We therefore assume that we have already classified all utterances and have with us a list of utterances with their classification results. To perform the second step in applying TAS we first need to make clear what this task of identifying and classifying relations really means: in *all* discussions each *utterance* (except from the first in the discussion) is related to an utterance *earlier* in the discussion. So each utterance (except from the first in the discussion) is a target node for which we have to search a source node. This we call the **identification** of a relation. Having found such a relation we have to name it, using one of the 8 relation-labels available in TAS: this we call the **classification** of a relation.

7.1 Our Approach

A relation in TAS can be described as a named arc between two nodes and can therefore be modeled as a tuple consisting of 3 elements: a source node, a target node and a relation label. To explain our approach let us assume that we want to identify a relation in which node Y is the target. Node Y can be any node except from the first node in the discussion. Our approach consists then of constructing a list of possible source nodes for node Y. For each node X in this list of possible source nodes features are extracted and used to compute the likelihood of node X being the actual source node for in a relation where node Y is the target node. Having selected the most likely node we then try to classify the relation i.e. select the right label for it.

7.2 Identifying Relations

Identifying a relation starts with constructing a group of possible source nodes. Such a group of possible source nodes consists of each utterance that fulfills the following

two constraints

- the possible source utterance has to be started before the target utterance
- the possible source utterance can not be on a closed branch (as described in chapter 2)

Having this group of possible source nodes we want to select the node which has the highest likelihood of being the source node in a relation with our current (target) node. To do this we will extract features from our target node and all possible source nodes. These features we then compare to features extracted from all source-target node combinations in our training set and we will use this information to calculate the likelihood of being a source node for all our possible source nodes. In the next sections we will describe the information we use to perform our likelihood-calculations.

In chapter 3 we have described the features used for classifying utterances. All these features (*Sentence Length*, *? and 'OR'*, *Last Label*, *Ngram Points* and *POS ngram Points*) except from *Last Label* are directly related to and can directly be derived from an utterance. These features turned out to be reasonably good features for classifying nodes, but for identifying relations we have developed other features, which will be described in the next sections.

7.2.1 Non-Conditional Chance

Each node or utterance has a label describing the type of the node and thus we can compute the chance that our source node will be of a possible type. For our total corpus for example table 2.3 shows us that the chance of a source node being of type *yes/no issue* is $\frac{729}{4202} \approx 0.17$. Based on the statistics obtained from our training set we thus can calculate the chance of a possible source node being the actual source node based on its type.

7.2.2 Conditional Chance

Just as we can compute non-conditional chance for a node being a source for a target node we can compute the conditional chance as well. Using the same table 2.3 we can compute that given the fact that a target node is of type *statement*, the chance of a source node being of type *open issue* is $\frac{305}{3518} \approx 0.09$.

7.2.3 Starttime Difference

Each utterance is annotated with its 'starttime', the time when a speaker started his utterance. It is expectable that over time the chance of an utterance being a source node of the current node decreases: referring back to something said 30 seconds ago is more likely then referring back to something said 30 minutes ago. This idea is not only inspired by our common sense, but by the TAS annotation manual [Rienks and Verbree, 2006] as well: *"in principle every next contribution of a participant becomes a child of the previous contribution, unless the current contribution relates more strongly to an ancestor of the previous contribution"*. The difference in starttimes for our target and possible source node could therefore be a very useful and easy computable feature.

7.2.4 Speaker

Where we would not expect that a fact like "Speaker X uttered sentence A" would influence the type of utterance A we do expect this information to be useful when searching for relations between utterances. With our common world knowledge it is fairly reasonable to expect that the chance of a *yes/no question* asked by speaker *P* answered by speaker *P* as well is smaller than the same question answered by another speaker. Table 7.1 shows statistics on *speaker change* on the HUB corpus. It shows that of the 600 relations with a source node labeled *yes/no Issue* and a target node labeled *Statement* 98 of the source and target nodes have the same speaker and 502 have different speakers. Using such statistics based on the training set we could compute the chance of a possible source node being the actual source node based on speaker information.

Source / Target	*Open Issue*	*A/B Issue*	*Yes/No Issue*	*Statement*	*Weak Statement*	*Total*
Open Issue	(15/14)	(5/7)	(15/26)	(45/260)	(5/21)	(85/328)
A/B Issue	(1/1)	(0/0)	(4/6)	(20/66)	(3/3)	(28/76)
Yes/No Issue	(8/16)	(4/1)	(27/27)	(98/502)	(7/39)	(144/585)
Statement	(20/66)	(13/15)	(49/152)	(396/1951)	(18/59)	(496/2243)
Weak Statement	(1/3)	(2/0)	(3/9)	(29/151)	(7/12)	(42/175)
Total	(45/100)	(24/23)	(98/220)	(588/2930)	(40/134)	(795/3407)

Table 7.1: Speaker information on relations (source and target by same speaker/by different speaker)

7.2.5 Ngram Combinations

Just as we used the ngrams of words to classify utterances we have chosen the ngrams of words to perform as a feature now as well. By computing all ngram combinations (n = 1 to 4) of the source and target nodes in our training set we hope to find ngram combinations which cue for a relation. An example of all the ngram combinations of a (possible) source node *"Do you think ?"* and a target node *"yeah ."* are presented in table 7.2. As this table shows, constructing ngram combinations easy results in a very large list. Since each ngram in our source node can be combined with each ngram in a target node. For two utterances with x and y words, with $x > 4$ and $y > 4$ we will find $(x+x-1+x-2+x-3)\cdot(y+y-1+y-2+y-3) = (4\cdot x-6)\cdot(4\cdot y-6)$ ngram combinations. Using these statistics based on the training set every possible source node can be attributed points based on the ngram combinations it has together with the target node. Still we probably would want to develop some sort of ngram combination selection method just as we developed an ngram selection method in our classification of the utterances.

All the ngram combinations are computed because it is reasonable to expect that some combinations of ngrams might cue for a relation especially for source and target nodes far apart. Let us for example look at figure 2.1, there we see a node with the following information *"P3: And solar cells, I dunno about that."* As explained in chapter 2 in principle each node relates to the previous one, unless it relates more strongly to an earlier ancestor. In this case the source node is a node which has appeared far earlier in the discussion. Looking for a lexical cue for this combination we see that in botch cases the word "solar" appears which cues us for a possible relation. But ngrams for a possible source and target node do not have to be exactly the same to

cue for a relation. Combinations of ngrams such as *"do you"* and *"yes"* are also very likely to cue for the presence of a relation. Using this feature we hope to find all sorts of ngram combinations which may cue for a relation. Still, one should be alert for possible overfitting.

Source node	**Target node**	**Source node**	**Target node**	**Source node**	**Target node**
Do	yeah	Do	.	Do	yeah .
you	yeah	you	.	you	yeah .
think	yeah	think	.	think	yeah .
?	yeah	?	.	?	yeah .
Do you	yeah	Do you	.	Do you	yeah .
you think	yeah	you think	.	you think	yeah .
think ?	yeah	think ?	.	think ?	yeah .
Do you think	yeah	Do you think	.	Do you think	yeah .
you think ?	yeah	you think ?	.	you think ?	yeah .
Do you think ?	yeah	Do you think ?	.	Do you think	yeah .

Table 7.2: ngram combinations for source node *"Do you think ?"* and target node *"yeah . "*

7.2.6 POS Ngram Combinations

English grammar does not only restrict us in making utterances, but it sometimes also helps us getting the underlying semantics of it. Let us illustrate this by a simple example. If we for instance would encounter an utterance as "No, he doesn't live there!" then it seems rather awkward that this utterance would be an answer to a question as "Do you live there?" Grammatical cues for this lay in the fact that our first utterance speaks in third person were our second speaks of the second person. Such grammatical cues can partly be obtained by POS-tagging our utterances. Just as we did with the words of an utterance we can than construct ngrams combinations, hoping that specific combinations turn out to be a good cue for the identification of a relation.

7.3 Classifying Relations

Having identified a relation between a source and target node we need to classify this relation i.e. labelling it with one of the labels mentioned in table 2.1. To do this we could make use of all the features described for nodes as well as for relations. The conditional chance attribute for instance could probably act very predictive for the labelling of a relation. Given the type of the source and target node the chance of the label of the relation can be looked up in the statistics acquired from the training set.

In general the task of classifying relations will be much alike classifying nodes. Each relation can only be labelled with one label and the information on which we base our classification is the same as used for the classification of nodes and identification of the relations. Actual research on the identification and classification of relations is not yet done by us.

Based on the figures presented in 2.4, showing the 'unbalancedness' of the labels used we might expect that special care needs to be addressed to the development of features distinguishing a relation from being *positive* or not.

Chapter 8

Dialogue Act Tagging

8.1 Introduction

Our task of classifying the HUB Corpus using TAS shows very much resemblance with Dialogue Act Tagging (DA Tagging). Dialogue Act Tagging stems from the work of the philosopher John Searle on speech acts [Searle, 1969]. Searle identified *locutionary* acts as well as *illocutionary* acts. In a conversation the locutionary acts involve creating, sound and thus words and (using a certain grammar) phrases with a certain sense, where the illocutionary act expresses the communicative force of the utterance. A speaker may ask a question or can inform someone.
At first Searle and other philosophers like Austin were most interested in *performative sentences* such as "I hereby declare war to X" or "I name this ship the Queen Elizabeth". But not only such utterances have an illocutionary force since an utterance as "I see a man wearing a green shirt" could easily be paraphrased as "I hereby asses that I see man wearing a green shirt" which reveals the illocutionary force of the first utterance, namely *asserting*.

This philosophy of speech acts is the philosophical basis of dialog acts. In dialog systems dialogue acts are used to assign an illocutionary force to an utterance. The task of assigning a dialog act out of a list of possible dialog acts to an utterance is called Dialogue Act Tagging.

This possible list of DA's, called DA schemes, come in various formats, although they also have lots of resemblances. Dialog acts as ASSERT, SUGGEST, ACCEPT and QUESTION appear in almost any scheme. The DAMSL architecture by [Core and Allen, 1997] is one of the first schemes. An extension of the DAMSL scheme, SWBD-DAMSL was used to tag the Switchboard Corpus (a large corpus of telephone conversations mentioned in section 8.4) and several other DA schemes are in use as well.

Just as in classifying utterances according to TAS, Dialogue Act Tagging is about assigning semantic labels to utterances on basis of syntactic cues. This syntax/semantics resemblance on utterance basis has been the reason why we did not only use the developed methodology for tagging the HUB Corpus with TAS, but also other corpora with dialog acts.

This chapter will further describe the dialogue act tagging of three corpora, namely the ICSI Meeting Corpus, the Switchboard corpus and the AMI Corpus.

8.2 Features Used in Previous Work

To tag an utterance with a DA several features can be used, just as we did to classify our utterances using TAS. One of the most used features are cue phrases, according to Jurafsky et al. [1998] because of its predictevness, or to cite him "Indeed, we and our colleagues as well as many other researchers have found that the words and phrases in a DA were the strongest cue to its identity". But although cue phrases probably are the most distinctive features, other features as sentence length and previous DA's can be very important as well. Tables 8.1 and 8.2 together give an overview of the features used in seventeen different papers on Dialog act tagging and present a lot of work done in DA tagging in a nutshell.

(1)	Ang et al. [2005]
(2)	Rosset and Lamel [2004]
(3)	Fernandez and Picard [2002]
(4)	Rotaru [2002]
(5)	Lendvai et al. [2003]
(6)	Andernach [1996]
(7)	Reithinger and Klesen [1997]
(8)	Venkataraman et al. [2002]
(9)	Keizer and Akker
(10)	Venkataraman et al. [2005]
(11)	Jurafsky et al. [1998]
(12)	Zimmermann et al. [2005a]
(13)	Zimmermann et al. [2005b]
(14)	Warnke et al. [1997]
(15)	Katrenko [2004]
(16)	Webb et al. [2005]
(17)	Stolcke et al. [2000]

Table 8.1: Papers used in table 8.2

Feature / Article	(1)	(2]	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)
Sentence length	X								X							X	
First two words	X	X															
Last two words	X																
First word of next segment	X																
Speaker		X															X
Number of utterances		X															
Prosodic			X		X						X			X			X
(Correct) last 10 DA's					X												
Words in last 10 DA's					X												
Utterance type						X											
Presence/absence 'Wh'-words						X											
Subject Type						X											
Specific cue phrases						X					X				X		
First verb type						X											
Second verb type						X											
Question mark						X											
Polygrams* of words in segment							X							X			
Ngrams of words in segment				X				X		X		X	X			X	X
Ngrams of previous DA's								X		X			X				X
Specific patterns									X								
Previous DA									X								
Next DA									X								
Grammar pattern									X		X						

Table 8.2: Features used as found in literature

*A polygram is a n-gram with varying size of n. It is superior to standard n-gram models because n can be chosen arbitrarily large and the probabilities of higher order n-grams are interpolated by lower order ones. [Warnke et al., 1997]

8.3 ICSI Meeting Corpus

The ICSI Meeting Corpus includes 75 naturally occurring meetings containing roughly 72 hours of multiparty speech data and associated human-generated word-level transcripts [Janin et al., 2003] and was hand-annotated for dialog acts [Dhillon et al., 2004, Shriberg et al., 2004]. The Meeting Recorder Dialog Act tag set (MRDA) uses 11 general tags and 39 specific tags. Each annotation requires one general tag and may be appended by a variable number of specific tags. The MRDA contains several classmaps as well. Using a classmap the total number of dialog acts can be mapped onto (far less) groups. For our experiments we have used a classmap which maps the DA's onto 5 distinct classes: statements (S), questions (Q), backchannels (B), fillers (F) and disruptions (D). Utterances that have not been annotated are labeled (Z). Finally the ICSI Corpus comes with a proposed train/test split. This split consists of 51 meetings (almost 80.000 utterances) which can be used for training, 11 meetings (around 13.500 utterances) for development, and 11 meetings (over 15.000 utterances) for evaluation. The 2 meetings which are left over are excluded because of their different nature.

The classification we performed on the ICSI corpus had the same outline as the classification of the HUB corpus. Since previous research [Ang et al., 2005, Zimmermann et al., 2005a,b] used the train/test split as proposed by ICSI we chose to use this same split in order to compare our results. In this previous research utterances of the class 'Z' were not used in classification thus experimenting with a classification task into 5 distinct classes. In our research we did use utterances of the class 'Z'. No ngram features were selected for the utterances of class 'Z', but all utterances of class 'Z' were classified. By performing this procedure we hoped to be able to show once more the importance of a type specific language model in our classification.

The best performance for DA classification based on human segmentations of the ICSI corpus is by our knowledge scored by Ang et al. [2005]: 81.18%. Table 8.3 shows our results using the same settings as we did in TAS classifying the HUB corpus, together with two computed baselines. Our performance of 89.27% is compared to the score of Ang et al. [2005] best explained by our use of all interesting ngrams, where they have focussed on the first and last words in an utterance. Although these results can not be compared directly since our experiments used a 6 distinct classes where Ang et al. [2005] used 5, ours are much better.

Feature set and parameters chosen	**Performance**
L_P_W (Non-Order Specific) (Compressed Top 10)	87.84
L_P_W (Order Specific) (Individual Top 10)	87.97
QMT_ORT_L_P_W (Order Specific) (Compressed Top 10)	87.82
QMT_ORT_L_P_W (Order Specific) (Individual Top 10)	87.98
QMT_ORT_L_LL_P_W (Order Specific) (Compressed Top 10)	89.13
QMT_ORT_L_LL_P_W (Order Specific) (Individual Top 10)	89.27
Majority class	55.46
LIT	63.57

Table 8.3: DA-classification results on the ICSI Meeting corpus

Table 8.4 shows the confusion matrix for the QMT_ORT_L_LL_P_W (Order Specific) (Individual Top 10) setting. First of all we see that utterances of the class 'Z' are quite well classified. We expect this to be due to the fact that no specific ngrams

have been selected to cue for the class 'Z'. Furthermore we see that a lot of *backchannels* are misclassified as *statements*. Analyzing the ngrams selected which should cue for *backchannels* we see that an ngram *'ok .'* is in the top 10 of ngrams cueing for *backchannels*. This ngram has 421 occurrences in utterances of the class *backchannels*, but occurs even more (1857 times) in utterances of the class *statement*. This is not a single case, but occurs more often: 3 out of the 10 selected best-cueing bigrams for *backchannels* have more occurrences in utterances of the class *statement* than in *backchannels*. So even if an ngram is among the best-cueing ngrams for a specific class it might even cue more for another class. Finally we notice that a considerable amount of utterances of each class are classified as *backchannels*. A shallow investigation of these utterances show that these utterances in general have few words. The *Length* feature seems to be quite important in classifying *backchannels*, just as we saw in classifying the HUB corpus.

a	b	c	d	e	f	$<--$ classified as
1558	1	18	0	384	0	a = B
55	1869	131	125	60	4	b = D
133	100	990	0	91	3	c = F
0	12	0	1095	4	4	d = Q
471	2	14	19	8057	6	e = S
5	4	3	0	3	177	f = Z

Table 8.4: Confusion matrix of QMT_ORT_L_LL_P_W (Order Specific) (Individual Top 10) ICSI setting

8.4 Switchboard Corpus

The Switchboard Corpus is a corpus of human-human conversational speech by telephone as described in Godfrey et al. [1992]. We used the same subset of this Switchboard Corpus as Stolcke et al. [2000] consisting of over 210,000 utterances grouped in 1,155 conversations. Dialogue act annotations based on the SWBD-DAMSL tag set were available for all of these conversations [Jurafsky et al., 1997]. Using this SWBD-DAMSL tag set 220 DA's were used for annotating. 130 of these occurred less than 10 times each and therefore all tags were clustered into 43 larger classes: the regular 42 DA's including the "+" DA, which is an equivalent DA for DAMSL "Segment". For our experiments we used this clustering of which an overview can be found in table 8.5. Previous research on the DA classification of the Switchboard Corpus has been done by Rotaru [2002] and Stolcke et al. [2000]. Stolcke reports on a 71% accuracy using a trigram word model, Rotaru scores 72%. Unfortunately both use different fixed train/test splits, without mentioning which split used. Furthermore the methods described are not very clear. Stolcke reports on the use of trigrams, where Rotaru uses bigrams as a feature, but in both cases it is not clear if *ngram selection methods* are used to make a distinction in the *'cueing power'* of an ngram. Rotaru uses the *'+'* DA, but Stolcke leaves it out.

In our experiments we have set up a 10-fold cross validation using 43 DA's (so including the *'+'* DA and mapping the *'%-'* "DA on the *'%'* DA). This makes our results not *directly* comparable to those of Stolcke and Rotaru. Table 8.6 presents

the performances of our different feature/parameter combinations on the Switchboard Corpus, together with two computed baselines.

SWBD-DAMSL	SWBD	SWBD-DAMSL	SWBD
Statement-non-opinion	sd	Action-directive	ad
Acknowledge (Backchannel)	b	Collaborative Completion	^2
Statement-opinion	sv	Repeat-phrase	b^m
Agree/Accept	aa	Open-Question	qo
Abandoned or Turn-Exit	%-	Rhetorical-Questions	qh
Appreciation	ba	Hold before answer/agreement	^h
Yes-No-Question	qy	Reject	ar
Non-verbal	x	Negative non-no answers	ng,nn^e
Yes answers	ny	Signal-non-understanding	br
Conventional-closing	fc	Other answers	no
Uninterpretable	%	Conventional-opening	fp
Wh-Question	qw	Or-Clause	qrr
No answers	nn	Dispreferred answers	arp,nd
Response Acknowledgement	bk	3rd-party-talk	t3
Hedge	h	Offers, Options Commits	oo,cc,co
Declarative Yes-No-Question	qy^d	Self-talk	t1
Other	o,fo,bc,by,fw	Downplayer	bd
Backchannel in question form	bh	Maybe/Accept-part	aap/am
Quotation	^q	Tag-Question	^g
Summarize/reformulate	bf	Declarative Wh-Question	qw^d
Affirmative non-yes answers	na,ny^e	Apology	fa
Thanking	ft	Segment	+

Table 8.5: SWBD-DAMSL tags and their mapping onto SWBD tags

Because of the number of DA tags used and the size of the Switchboard Corpus we were not able to compute results for the classifiers using individual ngrams. Therefore we would need around 43 (Tags) $\times 4$ (Orders) $\times 10$ (Cue ngrams) $\times 2$ (POS-tags and 'normal' words) $= 3440$ features. For the same reason we were not able to compute a results for the LIT set. This once more shows the advantage of using a *'compressed'* feature set. Although, it must be said that our approach of computing performances for the Switchboard Corpus costed a lot of computing power, even for the compressed feature set. The process of part-of-speech tagging, ngramming and classifying all 10 folds created out of the 210,000 utterances available in the Switchboard Corpus took about 3 days for each classifier-setting.

Table 8.7 shows the accuracy and number of tokens per DA for our best results, using the $QMT\_ORT\_L\_LL\_P\_W$ (Order Specific) (Compressed Top 10) setting. These figures have been distilled from the confusion matrix as depicted in 8.8.

We can conclude that in general DA-classes having few tokens perform worse than classes having much tokens. An exception to this is the *'qy^d'* tag, which is a 'Declarative Yes-No-Question'. Declarative questions are in the annotation manual described as "utterances which function pragmatically as questions but which do not have "question form" [...] Declarative questions normally have no wh-word as the argument of the verb (except in "echo-question" format), and have "declarative" word order in

Feature set and parameters chosen	Performance
L_P_W (Order specific) (Individual Top 10)	uncomputable*
L_P_W (Non-Order Specific) (Compressed Top 10)	60.57
QMT_ORT_L_P_W (Order Specific) (Compressed Top 10)	60.22
QMT_ORT_L_P_W (Order Specific) (Individual Top 10)	uncomputable*
QMT_ORT_L_LL_P_W (Order Specific) (Compressed Top 10)	65.68
QMT_ORT_L_LL_P_W (Order Specific) (Individual Top 10)	uncomputable*
Majority class	33.73
LIT	uncomputable*

Table 8.6: DA-classification results on the switchboard corpus

**Using a machine with 2048MB internal memory, the classifier ran out of memory*

which the subject precedes the verb." [Jurafsky et al., 1997]. This description also explains why the majority of misclassifications of the *'qyˆd'* and *'qwˆd'* are classified as a 'Statement-non-opinion'.

The *'+'* Dialog Act tag is at the moment classified as an individual tag. This leads to an accuracy of 47.66% for this tag. Our performance might be improved by classifying this tag according to the DA-tag it 'continues'. Utterances are classified as *'+'* whenever the utterance actually is a continuation of an earlier utterance by the same speaker. Classifying such DA's by *relabeling* them with the DA of the utterance they continue might therefore improve performance. To get some understanding of the influence of the *'+'* tag we performed a 10 fold cross-validation experiment in which utterances of the class *'+'* were discarded, resulting in a classification accuracy of 70.26%. Although this result shows the influence of this *'+'* class, better practice would be to *relabel* all utterances of class *'+'* to the class of the utterance they continue.

Furthermore the confusion matrix shows once more the problem of the 'unbalancedness' of corpora. Classes with much occurrences, such as *'sd'* and *'sv'* 'draw' utterances of other classes to it. It might be worthwhile to investigate if such classes with much occurrences do need their 'own' cue ngrams. Not computing the cue ngrams for such classes might help classifying utterances 'belonging' to smaller classes.

Interesting to see is the performance on the class *'b'* (*Backchannel*). With an accuracy of 91.73% *backchannels* are correctly classified. Although this is encouraging, it must be remarked that over 25% of the utterances of *Agree/Accept* are also classified as *Backchannel*. So, although backchannels are classified rather accurately, a real distinction between *backchannels* and *agrees/accepts* is not made.

DA	Accuracy	Occurrences	Accuracy	DA	Occurrences
%	73.27	13928	aa	55.13	9944
nn	83.60	1262	aap/am	4.26	94
no	19.31	259	ad	9.65	684
ny	81.42	2734	ar	19.42	309
oo, cc, co	0.97	103	arp, nd	3.72	188
qh	4.65	516	b	91.73	34438
qo	43.73	574	bˆm	11.09	622
qrr	44.21	190	ba	55.57	4306
qw	52.45	1794	bd	36.47	85
qwˆd	0.00	79	bf	3.63	853
qy	56.43	4255	bh	69.09	961
qyˆd	7.30	1137	bk	51.91	1181
sd	73.42	67348	br	52.63	266
sv	35.72	23424	fa	15.49	71
t1	1.06	94	fc	86.05	2280
t3	20.48	83	o, fo, bc, by, fw	39.25	1098
x	92.81	3284	fp	71.50	207
+	47.66	16744	ft	4.29	70
ˆ2	6.44	652	h	61.06	1058
ˆg	40.45	89	na, nyˆe	35.29	765
ˆh	20.91	507	ng, ng, nnˆe	19.50	282
ˆq	15.10	894			

Table 8.7: Accuracy per DA on the QMT_ORT_L_LL_P_W (Order Specific) (Compressed Top 10) setting

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	aa	ab	ac	ad	ae	af	ag	ah	ai	aj	ak	al	am	an	ao	ap	aq	< −− classified as
10205	26	0	20	14	214	2	19	8	7	744	12	76	1	17	2	4	2	0	9	28	0	0	15	17	4	6	2	32	0	6	5	7	24	1	49	7	1431	459	1	0	18	434	a = percent
49	42	0	7	9	19	0	5	1	0	14	17	42	0	10	0	0	2	0	1	1	0	0	1	4	1	0	0	0	0	6	0	0	1	0	13	6	215	109	0	0	3	74	b = ^2
2	2	36	0	2	5	0	1	0	0	6	1	0	0	0	11	0	2	0	0	0	0	0	0	2	0	0	0	1	0	0	0	0	1	0	11	1	1	2	0	0	0	2	c = ^g
38	0	0	106	2	6	1	8	2	5	25	1	14	0	1	1	2	0	0	3	4	1	0	1	15	3	1	3	2	1	0	1	2	1	0	7	5	193	30	0	0	0	22	d = ^h
21	6	1	3	135	9	0	6	1	0	13	4	13	0	5	0	0	0	1	1	0	0	0	4	0	1	1	0	2	0	2	2	0	14	0	25	3	456	117	0	1	0	47	e = ^q
187	18	3	8	12	5482	8	10	21	2	2820	9	191	5	15	0	15	0	1	10	3	0	1	3	15	0	30	0	47	1	3	0	0	0	0	6	3	501	368	0	1	5	140	f = aa
5	0	0	0	0	19	4	1	1	1	4	1	6	0	0	0	1	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	23	13	0	0	0	11	g = aap
24	2	0	1	7	20	0	66	1	2	5	1	5	1	7	0	0	2	0	3	7	2	0	0	1	1	0	0	0	1	2	1	0	5	0	23	4	311	102	0	0	0	77	h = ad
12	2	0	1	2	93	1	1	60	1	11	0	6	2	1	0	0	0	0	0	3	0	0	2	0	0	28	0	1	0	0	0	0	0	0	2	0	49	20	0	0	2	9	i = ar
19	0	0	7	0	4	0	0	1	7	1	0	2	0	1	0	0	0	0	1	0	0	0	0	21	5	2	2	0	0	0	0	3	0	0	1	2	86	15	0	0	1	7	j = arp
541	13	1	12	7	1263	0	11	8	1	31589	28	165	1	6	7	256	2	1	10	19	2	0	3	14	1	12	0	141	1	3	1	0	1	0	2	4	128	21	0	0	21	142	k = b
23	20	0	2	5	26	0	2	1	0	186	69	48	0	6	0	4	4	0	0	9	2	1	0	7	0	10	0	4	0	1	0	0	3	0	11	2	113	24	0	0	14	25	l = b^m
72	27	0	10	9	410	5	6	5	0	220	29	2393	5	23	9	14	0	3	5	6	3	1	4	6	3	1	1	2	0	2	2	0	4	1	12	8	522	282	0	0	41	160	m = ba
1	0	0	0	0	7	0	0	2	0	6	0	7	31	2	1	2	0	1	4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	10	6	0	0	0	3	n = bd
31	13	0	0	6	15	0	4	0	1	9	14	35	1	31	2	0	1	0	2	4	0	0	2	2	0	0	1	0	0	1	0	0	4	0	27	9	355	165	0	0	2	116	o = bf
4	2	4	1	2	10	1	1	0	0	95	0	16	0	3	664	2	5	0	0	2	0	0	0	1	0	0	0	0	0	2	1	0	13	1	86	4	17	9	0	0	0	15	p = bh
13	1	0	2	0	21	0	0	1	0	462	4	19	0	3	0	613	0	0	2	1	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1	24	0	0	1	5	5	q = bk
3	3	1	0	0	2	0	0	0	0	3	2	2	0	4	9	0	140	1	1	0	1	0	0	2	0	0	0	0	0	3	1	1	22	0	39	9	5	4	0	0	0	8	r = br
1	0	0	1	0	1	1	2	1	0	0	2	4	0	1	0	0	1	11	1	3	1	0	4	2	0	0	0	0	0	0	0	0	0	0	0	0	24	3	0	0	2	5	s = fa
23	0	0	0	1	6	1	5	0	4	32	1	12	4	1	2	9	1	0	1962	2	1	13	0	1	0	1	0	0	2	0	0	1	7	0	8	2	111	29	0	0	4	34	t = fc
43	0	1	2	1	18	0	7	5	1	57	4	19	1	2	0	3	0	1	2	431	7	0	3	3	0	8	1	2	0	0	1	0	3	0	11	2	97	24	0	1	8	329	u = fo
2	0	0	1	0	1	0	4	0	0	6	0	2	0	0	1	0	0	1	1	4	148	2	0	0	0	0	0	0	0	0	0	0	7	0	2	0	21	2	0	0	0	2	v = fp
0	0	0	0	0	5	0	1	0	0	2	1	5	0	1	0	2	0	0	28	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14	3	0	0	0	5	w = ft
35	2	0	3	5	7	0	1	1	0	3	0	7	0	0	1	0	0	1	0	2	0	0	646	3	3	0	11	1	0	1	0	0	2	0	1	2	244	46	1	1	0	28	x = h
14	1	1	21	0	23	1	2	0	14	27	6	9	0	5	0	2	1	1	1	5	0	0	2	270	40	3	21	19	0	1	0	2	0	0	7	9	188	34	0	0	1	34	y = na
6	1	0	4	0	2	0	0	1	7	1	1	2	0	1	0	0	2	0	1	0	0	0	2	38	55	3	10	3	0	0	0	0	1	0	3	2	108	17	0	0	0	11	z = ng
13	2	0	0	1	92	0	0	28	0	24	1	5	1	0	0	0	0	0	0	2	0	0	0	5	3	1055	0	4	0	0	0	0	0	0	0	0	20	1	0	0	0	5	aa = nn
9	0	0	6	0	2	0	1	1	9	2	1	2	0	0	0	0	0	0	0	0	0	0	33	31	10	0	50	0	0	0	0	0	2	0	0	1	74	13	0	0	0	12	ab = no
43	0	2	3	0	69	2	0	0	2	323	1	1	0	0	0	0	0	0	0	2	0	0	0	18	0	3	1	2226	0	1	0	0	0	0	1	0	19	2	0	0	4	11	ac = ny
0	1	0	4	2	4	0	2	0	0	0	0	3	0	2	0	0	0	0	2	0	0	1	0	1	0	0	0	0	1	2	0	0	0	0	2	0	59	11	0	0	0	6	ad = oo
17	3	1	0	8	6	0	5	0	0	4	2	6	0	4	6	0	4	0	0	3	0	1	3	1	0	0	0	0	0	24	11	0	42	0	82	4	142	91	0	0	0	46	ae = qh
9	0	0	0	3	3	0	3	0	0	1	3	2	0	2	4	0	2	0	0	0	1	0	1	0	0	0	0	0	0	11	251	1	152	0	52	4	47	7	0	0	0	15	af = qo
8	0	0	1	0	0	0	1	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	0	2	0	0	2	0	0	3	3	84	9	0	50	0	14	2	0	0	0	7	ag = qrr
42	8	0	2	16	3	0	4	0	1	4	2	4	0	2	7	0	32	0	1	2	5	0	0	1	1	1	1	0	3	31	98	5	941	2	307	18	134	37	5	2	1	71	ah = qw
1	0	0	1	0	2	0	1	0	0	0	0	1	0	0	1	0	1	0	0	1	1	0	3	0	0	0	0	0	0	1	0	0	6	0	3	2	42	9	0	0	0	3	ai = qw^d
115	17	10	13	20	13	0	27	0	3	16	11	14	0	27	130	0	41	1	3	7	4	0	0	11	4	3	4	9	1	66	40	35	244	5	2401	100	495	175	1	1	1	187	aj = qy
17	9	1	3	6	9	1	8	0	1	6	5	14	1	15	18	0	19	0	1	2	0	0	5	14	4	1	2	0	0	4	8	3	44	1	153	83	482	116	0	0	2	79	ak = qy^d
1596	164	1	149	231	542	21	235	31	71	128	72	507	10	240	13	16	4	9	81	70	12	7	187	178	81	13	45	9	34	94	18	13	86	15	305	219	49449	7847	5	12	50	4478	al = sd
611	135	0	39	113	479	14	102	11	23	58	23	380	6	136	4	1	0	3	38	23	5	4	39	34	15	5	10	1	7	66	14	7	43	6	123	77	10446	8366	0	1	10	1946	am = sv
5	1	0	4	2	4	0	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	3	0	2	4	0	7	2	41	7	1	0	0	6	an = tl
4	0	0	1	0	2	0	3	0	1	1	0	1	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	4	0	4	1	32	5	0	17	0	4	ao = t3
2	5	0	1	0	8	0	3	0	0	14	3	25	0	2	0	0	0	0	6	4	1	1	1	1	0	2	0	2	0	0	0	0	0	0	0	1	50	12	0	0	3048	92	ap = x
662	66	3	24	35	195	6	57	8	7	148	39	167	3	91	16	7	14	6	69	23	0	3	37	55	10	6	4	5	2	36	14	5	75	2	188	57	4744	1821	0	4	49	7981	aq = +

Table 8.8: Confusion matrix of QMT_ORT_L_LL_P_W (Order Specific) (Individual Top 10) Switchboard setting

8.5 AMI Corpus

The AMI Corpus is a collection of 100 hours of meetings of groups of four people. All meetings are in English, but a large proportion of speakers are non-native English speakers. Transcriptions of all meetings are available as well as several *layers* of annotation [Carletta et al., 2005]. The corpus holds a total of 80 meetings into 20 series of 4 meetings, which have been recorded at the University of Edinburgh (United Kingdom), the IDIAP research Institute (Switzerland) and at TNO (the Netherlands). Since not all the dialog annotations of the meetings were yet available we had to suffice with a few meetings less then available in the total corpus. Our corpus consisted of 76 of the available 80 meetings, which was good for over 50,000 utterances. The DA tag set consisted of 15 tags as presented in table 8.9.

Class	**Tag**
Backchannel	bck
Stall	stl
Fragment	fra
Inform	inf
Elicit-Inform	el.inf
Suggest	sug
Offer	off
Elicit-Offer-Or-Suggestion	el.sug
Assess	ass
Elicit-Assessment	el.ass
Comment-About-Understanding	und
Elicit-Comment-About-Understanding	el.und
Be-Positive	be.pos
Be-Negative	be.neg
Other	oth

Table 8.9: Dialog Acts used for the AMI HUB corpus

The classification we performed on the AMI corpus had the same outline as the classification of the HUB corpus. Unfortunately we were unable to make use of the *Last Label* feature because at the time not all the meeting were annotated with dialog acts with timing information. Since the AMI Corpus is a new corpus no previous DA tagging results have been published yet, so we unluckily have no performances to compare our results to.

To research the influence of the size of the language model we performed a 4-fold and a 10-fold experiment. In the 4-fold experiment $\frac{3}{4}$ of the training set can be used to construct a language model, where as $\frac{9}{10}$ of the training set can be used in case of our 10-fold experiments. Our results show that this increase of the size of the training set improves performance.

Table 8.10 shows our results for the DA classification of the AMI corpus. Regardless of the number of folds we used for cross-validation our classifier performed best using the L_1234P_1234W (Order Specific) (Individual Top 10) setting. Tables 8.11 and 8.12 show the confusion matrices for this setting, using respectively 4 and 10 folds for cross-validation. Interesting to see is that almost all classes (except from *el.inf* and *inf*)benefit from the increased size of the training set using 10 folds instead of 4. A

Feature set and parameters chosen	**4 fold**	**10 fold**
QMT_ORT_L_P_W (Order Specific) (Compressed Top 10)	53.49	53.52
QMT_ORT_L_P_W (Order Specific) (Individual Top 10)	57.84	57.82
L_P_W (Order Specific) (Individual Top 10)	57.46	57.94
L_P_W (Non-Order Specific) (Individual Top 10)	52.86	53.24
Majority class	28.69	28.69
LIT	43.87	44.24

Table 8.10: DA-classification results on the AMI corpus

comparison of the classification performances for each class is shown in table 8.13. A deeper analysis shows that the ngrams selected in both setups are very much alike.

The size of the training sets influences the classifiers performance in two ways: first by selecting the right ngrams which will work as a feature and second in learning the classifier. The 20% incremented size of the training hardly affected the selected ngrams, but did help to train the classifier better for almost any class, regardless of the number on instances occurring in the class.

Both confusion matrices show that there is not a specific class for which better features need to be found. There is an overall confusion for almost each class, therefor better performance might only be achieved by developing better ways of selecting cue ngrams.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	$<--$ classified as
4339	2281	3	77	44	66	6	5	252	2756	47	60	339	324	72	a = ass
799	5385	0	5	9	3	0	0	58	62	0	38	106	1	37	b = bck
14	1	2	1	0	0	1	0	1	34	2	1	2	1	1	c = be.neg
238	31	3	385	16	38	2	0	29	363	12	14	15	58	7	d = be.pos
62	14	0	13	343	426	33	11	35	224	6	2	7	70	27	e = el.ass
86	6	1	16	272	1195	59	11	90	415	7	4	20	77	33	f = el.inf
16	1	1	4	32	145	51	0	17	77	3	1	3	20	1	g = el.sug
17	2	0	0	34	32	1	12	5	9	0	4	16	0	3	h = el.und
251	64	0	14	21	48	7	1	7007	596	17	13	361	89	6	i = fra
2066	391	7	121	90	206	32	3	582	12749	116	50	178	1163	26	j = inf
78	18	1	10	8	15	2	0	31	331	196	7	7	66	2	k = off
294	212	1	27	15	41	1	3	72	331	5	154	62	30	30	l = oth
885	423	0	10	5	12	1	6	630	202	4	27	2406	19	28	m = stl
467	15	1	46	81	134	21	1	210	2675	39	28	40	1263	6	n = sug
429	353	0	14	39	108	1	0	26	177	2	39	112	6	99	o = und

Table 8.11: The confusion matrix for the L_1234P_1234W (Order Specific) (Individual Top 10) setting on the AMI corpus, using 4 fold cross-validation

8.5.1 Automatic Speech Recognition

TAS is as described not only a good structure to help people understand discussions, but might as well offer a trustworthy representation of a meeting which can be used by for instance a virtual chairman. When used in a real-time environment, no use can be made of transcriptions, so we would like to know the classification performance on Automatic Speech Recognition (ASR) output. Unfortunately at the moment there were no TAS schemes available with start- *as well as* endtime annotations. These characteristics together were available for some DA annotations on AMI meetings. Therefore we have performed a DA classification experiment on a corpus of 10 AMI

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	$<--$ classified as
4534	2313	3	68	38	73	8	5	262	2699	40	75	261	309	62	a = ass
751	5461	0	7	9	5	0	0	58	72	1	39	109	2	31	b = bck
18	0	2	1	0	0	0	0	2	32	2	2	2	0	0	c = be.neg
239	15	2	428	10	37	2	0	26	361	9	11	14	52	11	d = be.pos
80	14	0	11	359	412	35	11	32	239	1	3	6	54	19	e = el.ass
72	7	0	26	291	1156	79	14	95	420	5	6	12	76	41	f = el.inf
17	1	1	3	48	130	53	0	15	80	2	0	4	22	4	g = el.sug
26	2	0	0	35	33	1	13	4	8	0	3	7	2	2	h = el.und
260	43	1	19	16	47	8	0	7100	558	16	16	341	98	3	i = fra
2179	353	5	126	84	199	22	3	589	12706	133	43	168	1225	19	j = inf
79	18	0	11	9	19	0	0	40	314	229	6	10	42	0	k = off
298	170	0	22	17	33	2	2	63	318	6	209	72	38	31	l = oth
888	402	1	12	10	13	1	6	647	192	3	22	2442	26	9	m = stl
455	12	0	43	72	129	23	0	225	2714	29	31	40	1273	7	n = sug
396	361	0	10	41	112	1	1	24	170	1	61	118	12	95	o = und

Table 8.12: The confusion matrix for the L_1234P_1234W (Order Specific) (Individual Top 10) setting on the AMI corpus, using 10 fold cross-validation

Setting	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
4 fold	40.66	82.81	3.28	31.79	26.94	52.14	13.71	8.89	82.48	71.70	25.39	12.05	51.65	25.12	7.05
10 fold	42.18	83.44	3.28	35.17	28.13	50.26	13.95	9.56	83.27	71.17	29.47	16.32	52.25	25.19	6.77

Table 8.13: A comparison of the classification performances for each for a 4 and 10 fold cross-validation on the AMI corpus

meetings, transcripted using ASR, consisting of 7496 utterances. Table 8.14 shows the performances scored on the DA classifying of this ASR corpus, together with two baselines.

Feature set and parameters chosen	**Performance**
QMT_ORT_L_P_W (Order Specific) (Compressed Top 10)	37.43
QMT_ORT_L_P_W (Order Specific) (Individual Top 10)	40.05
L_P_W (Order Specific) (Individual Top 10)	40.26
L_P_4W (Non-Order Specific) (Compressed Top 10)	37.05
Majority class	30.03
LIT	32.68

Table 8.14: DA-classification results on the AMI corpus

Table 8.15 depicts the confusion matrix for the L_1234P_1234W (INDIVIDUAL T10) setting. This matrix shows that most utterances are classified as *Assess (a), Backchannel (b), Fragment (i)* and *Inform (j)*. These classes all have over 750 occurrences in the corpus, where all utterances belonging to other classes occur less than 600 times. The fact that utterances of class X occur more often than utterances of class Y causes class X of having a bigger chance of selecting the best cueing ngrams than class Y. This together with the relative small size of this ASR AMI corpus explains the somewhat disappointing performance.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	$<--$ classified as
317	508	0	10	7	14	0	0	97	313	4	3	23	16	0	a = ass
32	1017	0	1	0	0	0	0	77	4	0	0	7	0	0	b = bck
1	0	0	0	0	0	0	0	1	1	0	0	1	0	0	c = be.neg
25	27	0	8	0	8	0	0	6	38	1	1	3	6	0	d = be.pos
14	23	0	2	6	11	0	0	15	68	1	0	2	8	0	e = el.ass
30	41	0	6	10	12	2	0	19	170	1	0	0	13	0	f = el.inf
4	3	0	0	1	4	1	0	1	25	0	0	0	5	0	g = el.sug
3	6	0	0	0	1	0	0	1	1	0	0	3	0	0	h = el.und
91	399	0	3	5	11	0	1	107	110	0	3	23	13	1	i = fra
210	364	2	16	29	43	7	1	70	1336	16	3	23	131	0	j = inf
9	20	1	1	1	0	0	0	4	60	1	0	3	4	0	k = off
21	54	0	1	1	0	0	0	21	27	0	0	0	1	0	l = oth
43	209	0	2	3	2	0	3	43	41	1	1	112	2	0	m = stl
42	90	0	3	3	21	7	0	20	301	2	1	4	101	0	n = sug
15	54	0	0	2	1	0	0	8	15	2	1	2	1	0	o = und

Table 8.15: The confusion matrix for the L_1234P_1234W (INDIVIDUAL T10) setting on the ASR AMI corpus

Chapter 9

Conclusion and Recommendations

In this report we have discussed the automatic classification of utterances according to TAS as well as given an approach on the identification and classification of relations. Features, classifiers and ngram selection methods have been discussed. It is now time to come to a conclusion on our work, to look back on what we have achieved and to see hat still has to be considered. In this chapter we will come to conclusions and present recommendations for further work.

9.1 Conclusion

The aim of our project has been to automatically classify utterance according to TAS. With an accuracy of 78.52% we can not see that we have succeeded, but we certainly have made a beginning. In our search of a good performing classifier we have developed interesting ngram features, using ngrams of words and parts-of-speech as well as in an individual as 'compressed' way. The acquisition of our features turned out to be a time consuming task, mostly because of the preprocessing tasks as part-of-speech tagging, ngramming and ngram selecting. This selection of the most predictive ngrams is in our believe one the most influencing factors of the performance of our classifier and we believe improvements still can be made.

But TAS is not only about utterances to classify, is is about relations to identify and label as well. Even though we did not have the time to perform experiments on this identification and classification of labels we have presented an approach of doing so, focusing on statistics obtained from available argument diagrams.

All and all, we believe that this report and our research is an interesting contribution to the field of utterance classification and argument diagramming. In the next sections we will discuss some specific issues as the automatic acquisition of cues, the compressed feature set and our dialog act tagging results.

9.1.1 Automatically Acquisition of Cues

In chapter 6 we discussed the LIT set [Samuel, 2000]. We used this LIT set as a baseline for our results. Using only automatically selected ngrams of words we obtained a score of 73.41% which is slightly better than our LIT-baseline of 71.79%. This score

increases to around 76% if we also make use of part-of-speech information. All and all this is another indication that automatically selected cue phrases perform relatively well.

9.1.2 Compact Feature Set

Our approach differs from other approaches in for instance dialogue act classifying (which show much resemblance to our task of classifying utterances) in the use of a *compressed* feature set. Unlike in other research like that of Ang et al. [2005] and Rosset and Lamel [2004] we have not only made use of the first two word in an utterance, but of each word. But unlike Rotaru [2002], Zimmermann et al. [2005b] and Warnke et al. [1997]. Our approach did not result in an extreme large feature set. In addition to this compaction we have made use of ngrams of POS-tags which has earlier been done in research concerning the creation of backchannels in a spoken dialogue system [Cathcart et al., 2003]. The *compaction* of our feature set does not only drastically decrease the size of our feature vector, but therefore does also decrease our computing time. This of course is by it self not an advantage, because success is not measured on computing time alone, but certainly is influenced by performance as well.

Therefore we have compared our 'compressed results' with results obtained by a classifier using the same ngrams and POS-ngrams as features, only not 'compressing' them, but using each (POS)ngram as an individual feature. Over different ngram selection methods and feature combinations (only word ngrams, only POS ngrams, word and POS ngrams combined, etc.) the increase of accuracy has been in the range of 1 to 3 percent. So by losing some accuracy computing time drastically decreases. The preprocessing time of obtaining the features of an utterance is hardly influenced by our choice of using a compressed or an individual feature set.

9.1.3 Dialog Act Tagging

One of the advantages of our methodology is the ability of using the same methodology for dialog act tagging. Our best performance of 89.13% on the ICSI corpus is the only slightly comparable result to other DA tagging classifiers. For the Switchboard corpus, earlier DA tagging results were available, but the absence of a well-defined train/test set made it impossible to compare these results with ours. We believe to be the first one performing a 10 fold cross-validation on this corpus and so our results can only be compared to the majority class baseline of 33.73%. Compared to this baseline our performance of 10000000 is a good improvement. The AMI corpus is a new corpus for which no results were available. Our best results of 57.94% has sat a more interesting baseline for further research, than the majority class baseline with a performance 28.69%.

9.2 Recommendations

As in almost any project time limits have seemed to be the hardest. Technologically much more things are possible to improve our results, but we have simply run out of time. In this section we will give our recommendations for future work on the TAS-classification of utterances.

9.2.1 Revision of the Corpus

In section 3.3 we spoke about 'ambiguous text in the transcriptions' of the HUB corpus. Having done all sorts of experiments on the corpus accidently few ambiguities and/or errors were found. Therefore a revision of (the annotations) of the HUB-corpus might be needed to correct these errors and improve non only the quality of the corpus, but the performance of our classifier as well.

9.2.2 Classifying Nodes and Relations at the Same Time

In chapter 4 we discussed our strategy of first classifying nodes and then identifying and classifying relations. An advantage of this approach is that each subtask can be approached like it exists without a context. But this at the same time is a disadvantage, because both tasks do exist in the context of each other.

Let us for example once more turn back to figure 2.1. At a given moment P0 utters *"There's also a watch moves around a great deal more ."* If we would classify this utterances together with identifying and classifying relations we would have the information of the TAS-diagram unto the time P0 utters what he has to say. We would signal that on the *yes/no issue* posted by P1 already two reactions, classified as *statement* have been given. Furthermore we would identify the speakers of those statements to be P2 and P3. All this information together makes it even more plausible that P0's utterance is a *statement* related to the *yes/no issue* posted by P1.

This example shows us that a combined classification of nodes and identification *and* classification of relations would give us more information on the context of an utterance and therefore would be a good possibility of improving classification performance.

9.2.3 Other Classifiers

We have used three classifiers available in Weka, namely J48, DecisionTable and MultilayerPerceptron. DecisionTable was chosen because of its good performance in earlier experiments, just as J48. J48 also had the advantage of being a rather fast classifier and is therefore a more interesting classifier when equal performances arise. A disadvantage of the J48 classifier is that it does not really *combine* several features in classification. As described in chapter 4 the J48 algorithm is trained by splitting up its data by each time using the feature which discriminates the data the best. The MultilayerPerceptron classifier on the other hand combines all the features available and uses a weighting system for its classification. Because of its '*system*' this MultilayerPerceptron has more opportunities of combining several features in its classification.

In our experiments we have done a shallow comparison of the classifiers mentioned, but a lot of other possible classifiers have not been used in the experiments. It would be an interesting extension to our work to use several other classifiers in our experiments, for example a classifier using a maximum entropy classification method which are much used in work by other researchers.

9.2.4 Assess Other Ngram-Selecting Methods

Our work has mostly concentrated on ngrams of words and POS-tags. Four general ngram-selecting methods have been used to perform several different experiments. The results of the experiments clearly show that for each classifier the ngram-selecting

method strongly influences the performance. We would therefore recommend to develop and research other ngram-selecting methods as well.

Our method for selecting the best 'cueing' ngrams for example might need to be revisited. Table 9.1 for example shows the 3th and 4th best ngrams that have been selected as cue phrases for the type *a/b issue*. Although the score for the ngram *', or'* (2.28) is better than the score for the ngram *'? three'* (0.75) it still is disputable whether *', or'* is a better cue for the type *a/b issue* than *'? three'*, since using *', or'* the types *statement* and *unknown* will profit more of the occurrence of the ngram than the type what is was meant for: *a/b issue*.

More research on scoring algorithms might results in better ngram selection methods and therefore a better performance on the classification task.

ngram	statement	weak statement	open issue	ab issue	yn issue	unknown
, or	47	4	4	14	4	13
? three	1	0	0	3	0	0

Table 9.1: 3th and 4th best ngram for type *a/b issue*

9.2.5 Ngram Points to Attribute

Not only the selection of the right ngrams influences the performance of our classifiers based on ngrams, but the points attributed to a feature when an ngram is present as well. In our research we have used the number of occurrences of an ngram as a feature value. In the case of classifiers based on individual ngrams, the number of times a specific ngram occurs in an utterance to classify is the value for the specific ngram-feature. In the case of a 'compressed' feature set the number of times an ngram occurs for each type in the training set is the value used for the ngram. Only in the case of the classifier based on normalized ngram points values, we have normalized these value over all ngrams in a specific type.

It might be worth to investigate other possible values one could assign to an ngram. One of the most obvious methods would be to assign the ngram the score it has acquired for each type by the TOPx ngram selection method. Another possibility is the number of occurrences for an ngram for a type normalized over all ngrams for that type. Finally we could think of a system in which we use the number of occurrences of an ngram, but group several numbers of occurrences. An example of such a system is depicted in 9.2.

Number of occurrences	Points
0 - 1	1
2 - 5	2
5 - 10	3
10+	4

Table 9.2: Example of grouped ngram occurrences values

9.2.6 Selecting Cue Ngrams Regardless of Their Type

In our experiments we have selected an equal amount of ngrams as a cue for each type. It is worthwhile researching the effects of selecting the n most predictive cue phrases regardless from the type they cue for. At the moment we have encountered situations in which the 8th best ngram for selecting type A (the ngram *'mm'* for type *unknown*) had a better score than the 2nd best ngram for selecting type B (the ngram *'?'* for type *A/B issue*). Still, when we would select the n best ngrams regardless for what type it cues for, it might be that the classification for some type will be far worse than others, because less cue ngrams might have been selected for that type. Further research has to show the consequences of such a setting.

9.2.7 Punctuation Features

The use of the presence or absence of a question mark might be a form of 'cheating', since in automatic speech recognition it is very hard to recognize whether an utterance is a question or not and thus deciding on placing a question mark in the output or not. Since we like to have our classification of a discussion using TAS to be applicable to discussion transcribed using automatic speech recognition we would like to get rid of this 'cheating'. But simply stop making use of the *QMT_ORT* feature is not a real solution to this 'cheating' problem, because among the selected ngrams several ngrams containing question marks, or in general punctuation might be present. So we would still make use of punctuation, which is very hard to recognize using automatic speech recognition. It would therefore be good to investigate the influence of such ngrams and the *QMT_ORT* feature to see if they really are needed for good classification

9.2.8 Attribute Evaluation

We have researched several features on which the utterances in a discussion could be classified using TAS. Our focus has mostly been on ngrams and ngram selection methods. We have shown the usefulness of combining several features and the settings of some parameters. Because computing power was not a real bottleneck in our research we have for instance used tri- and quadrigrams where it is disputable whether they really add something to the performance of the classifier when already using several other features. Therefore more research is necessary to figure out which features are more useful than others.

9.3 Final Thoughts

With this project a beginning is made in the automatic classification of utterances according to TAS, to, in the end, structure discussions in argument diagrams. We believe our work to be a valuable supplement to earlier research in the meeting and argumentation field. With projects as these we hope to come closer to applications as mentioned by Ellis et al. [2001]: virtual meeting agents assisting in and around meetings for improved effectiveness and efficiency of those meetings.

Appendix A

In this appendix which is taken from Weijden [2005] of the labels and their definitions are described in detail. First the table with descriptions of the unit labels (node labels) is shown and the second table contains the relation labels.

A.1 Unit Labels

Open Issue	Issues that are raised where every possible response could be a solution: 'Wh. questions *Example* : "What does it prove?"
A/B Issue	Issues that are raised where the possible responses are explicitly enumerated. *Example* : "What is the most intelligent animal: A, B, or C?"
Yes/No Issue	Issues that are raised where the possible responses are Yes and No. *Example* : "Are all those names fake?"
Statement	Claims without a weakening qualifier. *Example*: "I Heard about their football team"
Weak Statement	Claims with a weakening qualifier. *Example* : "Yeah, probably"

A.2 Relation Labels

Positive	This relations relates statements (child) to issues (parent) and issues to issues and statements to statements. The label is used if the child aims to support the parent. *Example* : The relation between the statement, "I vote for ants." the statement "Yeah, same."
Negative	This relations relates statements to issues, issues to issues and statements to statements. The label is used if the child aims to refute the parent. *Example* : The relation between the statement, I would bet on cow. and the statement "I would eliminate cow anyway."
Uncertain	This relations relates issues to statements. The label is used if it is unclear how, being it either positive or negative, the child (issue) relates to the parent (statement). *Example* : The relation between the statement, "There is a high degree of similarity." and the open issue "What about HD distance?"
Option	This relations relates statements to issues and statements to statements. The label is used if the child is a possible answer, option or solution with respect to the parent. *Example* : The relation between the open issue, "What is the capital of Moldova?" and the statement "I would say Chisinau"
Option Exclusion	This relations relates statements to issues and issues to issues. The label is used if the child excludes one or more possible answers, options or solutions with respect to the parent. *Example* : The relation between the open issue, "What is the most intelligent animal?" and the yes-no issue "I wouldn"t look at an ant as a brilliant individual, by itself it is nothing, right?"
Elaboration	This relations relates issues to issues. or issues to statements or statements to issues, or statements to statements. The label is used if the child repeats, or asks to repeat the parent in other wordings. It can also be used if additional information about the parent is asked or given. *Example* : The relation between the statement "Ants are able to modify our garden" and the open issue, "What do you mean by modifying the environment?"
Specialization	This relations relates statements to statements and issues or issues. The label is used if the child is a specialization of the parent, and the child has a different semantic meaning, but the children"s children still provide either support or refutation to the parent. *Example* : The relation between the open issue, "What is the capital of Moldova?" and the open issue "What sounds most Moldovian?"
Request	This relations relates statements to Yes/No or A/B or Open issues. The label is used if the child asks for more information about the parent. This can be information about the topic or asking to repeat the statement when someone did not hear it. *Example* : The relation between the statement, "Ants are able to modify our garden." and the and the Open issue "What do you mean by modifying the environment?"
Subject To	This relations relates statements to Yes/No or A/B issues or statements to statements. The label is used if the child points out criteria or dependencies that have to be fulfilled before the parent can be supported or refuted. *Example* : The relation between the statement, "If you leave something in the kitchen, you are less likely to find a cow there." and the statement "That depends if the cow is very hungry."

Bibliography

T. Andernach. A machine learning approach to the classification of dialogue utterances. *CoRR*, cmp-lg/9607022, 1996.

J. Ang, Y. Liu, and E. Shriberg. Automatic dialog act segmentation and classification in multiparty meetings. In *Proceedings of the 30th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2005.

S. Banerjee and T. Pedersen. The design, implementation, and use of the Ngram Statistic Package. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, February 2003.

J. Carletta. Assessing agreement on classification tasks: The kappa statistic. *Computational Linguistics*, 22(2):249–254, 1996.

J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karasaiskos, W. Kraaij, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner. The AMI meeting corpus: A pre-announcement. In *Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI)*, Edinburgh, 2005.

N. Cathcart, J. Carletta, and E. Klein. A shallow model of backchannel continuers in spoken dialogue. In *EACL '03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, pages 51–58, Morristown, NJ, USA, 2003. Association for Computational Linguistics. ISBN 1-333-56789-0.

J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46, 1960.

R. Cohen. Analyzing the structure of argumentative discourse. *Comput. Linguist.*, 13 (1-2):11–24, 1987. ISSN 0891-2017.

M. G. Core and J. F. Allen. Coding dialogues with the DAMSL annotation scheme. In David Traum, editor, *Working Notes: AAAI Fall Symposium on Communicative Action in Humans and Machines*, pages 28–35, Menlo Park, California, 1997. American Association for Artificial Intelligence.

R. Dhillon, S. Bhagat, H. Carvey, and E. Shriberg. Meeting recorder project: Dialog act labeling guide. Technical report, ICSI, February 2004.

F.H. Eemeren. A glance behind the scenes: The state of the art in the study of argumentation. *Studies in Communication Sciences*, 3(1):1–23, 2003.

C.A. Ellis, P. Barthelmess, B. Quan, and J. Wainer. Neem: An agent-based meeting augmentation system. Technical Report CU-CS-937-02, University of Colorado at Boulder,Department of Computer Science, 2001.

R. Fernandez and R.W. Picard. Dialog act classification from prosodic features using support vector machines. In *Proceedings of speech prosody 2002*, April 2002.

W. Francis. A tagged corpus - problems and prospects. pages 192–209, 1980.

B. Fraser. An approach to discourse markers. *Journal of Pragmatics*, pages 383–395, 1990.

M. Galley, K. McKeown, J. Hirschberg, and E. Shriberg. Identifying agreement and disagreeement in conversational speech: Use of bayesian networks to model pragmatic dependencies. Technical report, Columbia University, SRI International, 2004.

J Godfrey, E. Holliman, and J. McDaniel. Switchboard: Telephone speech corpus for research and development. In *Proceedings of the IEEE Conference on Acoustics, Speech, and Signal Processing*, volume 1, page 517520, San Francisco, March 1992.

B. J. Grosz and C. L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.

M. Halliday and R. Hasan. *Cohesion in English.* Longman, London, England, 1976.

P. Heeman, D. Byron, and J. Allen. Identifying discourse markers in spoken dialog., 1998.

J. Hirschberg and D. Litman. Empirical studies on the disambiguation of cue phrases. *Comput. Linguist.*, 19(3):501–530, 1993. ISSN 0891-2017.

A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. The ICSI meeting corpus. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 364–367, 2003.

D. Jurafsky, E. Shriberg, and D. Biaska. Switchboard SWBD-DAMSL shallow-discourse-function annotation (coders manual, draft 13). Technical report, Univ. of Colorado, Inst. of Cognitive Science, 1997.

D. Jurafsky, E. Shriberg, B. Fox, and T. Curl. Lexical, prosodic, and syntactic cues for dialog acts. In Manfred Stede, Leo Wanner, and Eduard Hovy, editors, *Discourse Relations and Discourse Markers: Proceedings of the Conference*, pages 114–120. Association for Computational Linguistics, Somerset, New Jersey, 1998.

G. Kanselaar, G. Erkens, J. Andriessen, M. Prangsma, A. Veerman, and J. Jaspers. *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making*, chapter Desiging Argumentation Tools for Collaborative Learning. Springer Verlag, London, UK., 2003.

S. Katrenko. Textual data categorization: back to the phrase-based representation. In *Proceedings in 2nd International IEEE Conference 'Intelligent systems', Vol. III*, pages 64–67, June 2004.

S. Keizer and R. op den Akker. Dialogue act recognition under uncertainty using bayesian networks. *Natural Language Engineering*. To appear.

A. Knott. *A Data-Driven Methodology for Motivating a Set of Coherence Relations*. PhD thesis, University of Edinburgh, Edinburgh, 1996.

Ron Kohavi. The power of decision tables. In *ECML '95: Proceedings of the 8th European Conference on Machine Learning*, pages 174–189, London, UK, 1995. Springer-Verlag. ISBN 3-540-59286-5.

W. Kunz and H.W.J. Rittel. Issues as elements of information systems. Working Paper WP-131, Univ. Stuttgart, Inst. Fuer Grundlagen der Planung, 1970.

P. Lendvai, A. van den Bosch, and E. Krahmer. Machine learning for shallow interpretation of user utterances in spoken dialogue systems. In *Proceedings of EACL-03 Workshop on Dialogue Systems: interaction, adaptation and styles of management*, pages 69–78, 2003.

W.C. Mann and S.A. Thompson. Rhetorical structure theory: A theory of text organization. Technical Report ISI/RS-87-190, University of Southern California, 1987.

D. Marcu. The rhetorical parsing of unrestricted natural language texts. In Philip R. Cohen and Wolfgang Wahlster, editors, *Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 96–103, Somerset, New Jersey, 1997. Association for Computational Linguistics.

M. Marcus, B. Santorini, and M. Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330, 1993.

M. Minsky and S. Papert. *Perceptrons*. MIT Press, 1969.

J. Moore, M. Kronenthal, and S. Ashby. Guidelines for AMI speech transcriptions. Technical report, IDIAP, Univ. of Edinburgh, February 2005.

A. Neass. *Communication and argument. Elements of applied semantics.* George Allen & Uwin Press, 1966.

J. Quinlan. *C4.5 : programs for machine learning*. Morgan Kaufmann, San Mateo, CA, USA, 1993. ISBN 1558602380.

R. Reichman. *Getting computers to talk like you and me*. MIT Press, Cambridge, MA, USA, 1985. ISBN 0-262-18118-5.

N. Reithinger and M. Klesen. Dialogue act classification using language models. In *Proceedings of EuroSpeech-97*, pages 2235–2238, 1997.

R. Rienks, D. Heylen, and E. van der Weijden. Argument diagramming of meeting conversations. Technical report, Universiteit Twente - Human Media Interaction, 2005a.

R. Rienks, A. Nijholt, and D. Reidsma. *Meetings and Meeting support in ambient intelligence*, chapter in Ambient Intelligence, Wireless Networking, Ubiquitous Computing. Artech House, Norwood, MA, USA, 2005b. In Press.

R.J. Rienks and D. Heylen. Automatic dominance detection in meetings using easily obtainable features. In *Revised Selected Papers of the 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms*, volume 3869 / 2006, pages 76–86, Endinburgh, Scotland, oct 2006. Springer-Verlag GmbH. ISBN=3-540-32549-2, ISSN=0302-9743.

R.J. Rienks and A.T. Verbree. *Twente Argument Schema Annotation Manual*. University of Twente, Enschede, the Netherlands, 0.6 edition, may 2006.

S. Rosset and L. Lamel. Automatic detection of dialog acts based on multi-level information. In *icslp*, pages 540–543, Jeju Island, October 2004.

M. Rotaru. Dialog act tagging using memory-based learning. 2002.

K. Samuel. *Discourse Learning: An Investigation of Dialogue Act Tagging using Transformation-Based Learning.* PhD thesis, Department of Computer and Information Sciences. University of Delaware. Newark, Delaware., 2000.

D. Schiffrin. *Discourse Markers*. Cambridge University Press., London, England, 1987.

D. Schum and A. Martin. Formal and empirical research on cascaded inference in jurisprudence. *Law and Society Review*, 17(1):105–152, 1982.

J. Searle. *Speech Acts: An Essa in the Philospohy of Language*. Cambridege University Press, 1969.

E. Shriberg, R Dhillon, S. Bhagat, J. Ang, and H. Carvey. The ICSI meeting recorder dialog act (mrda) corpus. In *Proc. HLT-NAACL SIGDIAL Workshop, Boston, April-May*, 2004. AMI-0.

A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. Van Ess-Dykema, and M. Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech, 2000.

S. Toulmin. *The uses of argument*. Cambridge University Press, 1958.

K. Toutanova, D. Klein, and C. Manning. Feature-rich part-of-speech tagging with a cyclic dependency network. 2003.

F. Van Eemeren, R. Grootendorst, and F. Snoeck Henkemans. *Argumentation*. Lawrence Erlbaum Associates, 2002.

A Veerman. *Computer-supported collaborative learning through argumentation*. PhD thesis, University of Utrecht, 2000.

A. Venkataraman, A. Stolcke, and E. Shirberg. Automatic dialog act labeling with minimal supervision. In *Proceedings of the 9th Australian International Conference on Speech Science & Technology*, December 2002.

A. Venkataraman, Y. Liu, E. Shriberg, and Stolcke A. Does active learning help automatic dialog act taggin in meeting data. In *Proc. Eurospeech*, 2005.

D. Walton. *Argument Structure: A Pragmatic Theory*, chapter 2.3 and 2.4, pages 49–56. University of Toronto Press, 1996.

R. Warner. *Discourse Connectives in English.* Garland Publications, New York, New York, 1985.

V. Warnke, R. Kompe, H. Niemann, and E Noth. Integrated dialog act segmentation and classification using prosodic features and language models. pages 207–210, September 1997. Eurospeech.

N. Webb, M. Hepple, and Y. Wilks. Dialogue act classification based on intra-utterance features, 2005.

E. van der Weijden. Participating in virtual meetings: Visualizing the design rationale. Master's thesis, Universiteit Twente, Enschede, the Netherlands, novmeber 2005.

P. Werbos. *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. PhD thesis, 1974.

I. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. The Morgan Kaufmann Series in Data Management Systems, Jim Gray, Series Editor, October 1999.

J. Yoshimi. The structure of debate. Technical report, University of Claifornia, Merced, September 2004.

M. Zimmermann, Y. Liu, E. Shriberg, and A. Stolcke. Toward joint segmentation and classification of dialog acts in multiparty meetings. In *Proceedings of MLMI*, 2005a.

M. Zimmermann, Y. Liu, E. Shriberg, and A. Stolcke. A* based joint segmentation and classification of dialog acts in multiparty meetings. In *Proceedings of IEEE Speech Recognition and Understanding Workshop*, 2005b.

I. Zukerman and J. Pearl. Comprehension-driven generation of meta-technical utterances in math tutoring. In *AAAI*, pages 606–611, 1986.

Index