About the Usefulness and Learnability of Argument-Diagrams from Real Discussions

Rutger Rienks and Daan Verbree

Human Media Interaction (HMI)
University of Twente, Enschede, The Netherlands
{Rienks, Verbree}@ewi.utwente.nl,
Home page: http://hmi.ewi.utwente.nl/

Abstract. This paper continues the work described in Rienks and Heylen [2005] about argument diagramming of meeting discussions. In this paper we introduce the corpus that we created, discuss a user experiment about the usability of the technique, and show that the units of the diagramming method (segmented user utterances) can be learnt and predicted with an accuracy of 88.4% and 82.2% on an unbalanced and balanced set respectively.

1 INTRODUCTION

Argumentation has been regarded as our primary means of making progress [van Gelder, 2002]. It is pervasive in everyday life and plays an important role in human communication. Argumentation is inherently related to discussions, conversations and meetings, the arenas where one argues with another and one or more sides are attempting to win the approval of the opponent or of a designated audience.

Within organizations the general visible results of conversations or meetings are normally nothing more than what one is able to recall, if lucky from some notes that were taken, or perhaps some more formal meeting minutes or a list of action items. Generally, a lot of energy and information that has been put into the actual outcome is never seen again.

In Twente we have tried to find an approach that is able to capture the lines of the deliberated arguments in meeting discussions. This approach, the TAS-schema, was introduced in Rienks and Heylen [2005] and promised to be a valuable technique for capturing organizational memory. The structure that the arguments encapsulate reveals information about the trail or path that has been taken and can show the line of reasoning at specific moments in time. The method can aid querying systems and can be used in meeting browsers (See fig 1). The possibility of preserving the arguments and their coherence relations for future explorations make them potentially valuable documents containing a tacit representation of otherwise volatile knowledge [Shum, 1997, Pallotta et al., 2005].

In this paper we show how we continued our research in this area. Before we elaborate on how we created a corpus of annotations in Section 3, Section 2 will

shortly re-introduce the TAS-schema. To validate the potential benefit of the schema, we conducted a user experiment to find out how useful the diagrams are in relation to other representations of the discussion when answering questions about them. The results of this experiment are described in Section 4. We conclude the paper with on-going work related to the learnability of (a subset of) the schema in Section 5 for e.g. an automatic tagger that hopefully one day can produce the actual schemes autonomously.

2 The Twente Argument Schema

The Twente Argument Schema (TAS) is a schema designed to create argument diagrams from meeting discussion transcripts. Following most of the diagramming techniques studied, application of the method results in a tree structure with labelled nodes and edges. The nodes of the tree contain parts of, or even complete, speaker turns whereas the edges represent the type of relation between the nodes.

In short TAS accounts for capturing the most important conversational moves in dialogues where participants discuss the pros and cons of certain solutions to a problem, providing arguments in favor of or against the various solutions. TAS distinguishes acts in which issues are raised (questions put forward) and statements for a position that are made. It allows one to indicate whether a statement is strong or weak. Whether statements agree or disagree with each other can be marked in the relations. In many cases statements are not simply in favor or against but variations of each other: restatements, specializations or generalizations.

TAS was constructed in a way that it preserves the conversational flow. 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. This was realized by assuring that in principle every next contribution of a participant becomes a child of the previous contribution, unless the current contribution relates more strongly to the parent of the previous contribution. An example of a resulting argument diagram is shown in Figure 1. For more information on the schema the reader is referred to Rienks and Heylen [2005] and Van der Weijden [2005].

3 Creating a corpus of Meeting Discussions

In order to realize a corpus of TAS-annotations both an annotation tool and a corpus of meetings was required.

The annotation tool, *ArgumentA*, was created by building further on a number of components described in Reidsma et al. [2005]. ArgumentA allows annotators to select text on a transcripion-view pane and label them similarly to dialogue-acts. The label is assigned by selecting the unit text with the mouse from the transcription pane and then pressing a button popping up a label selection window from which the unit label can be picked. The labelled units appear

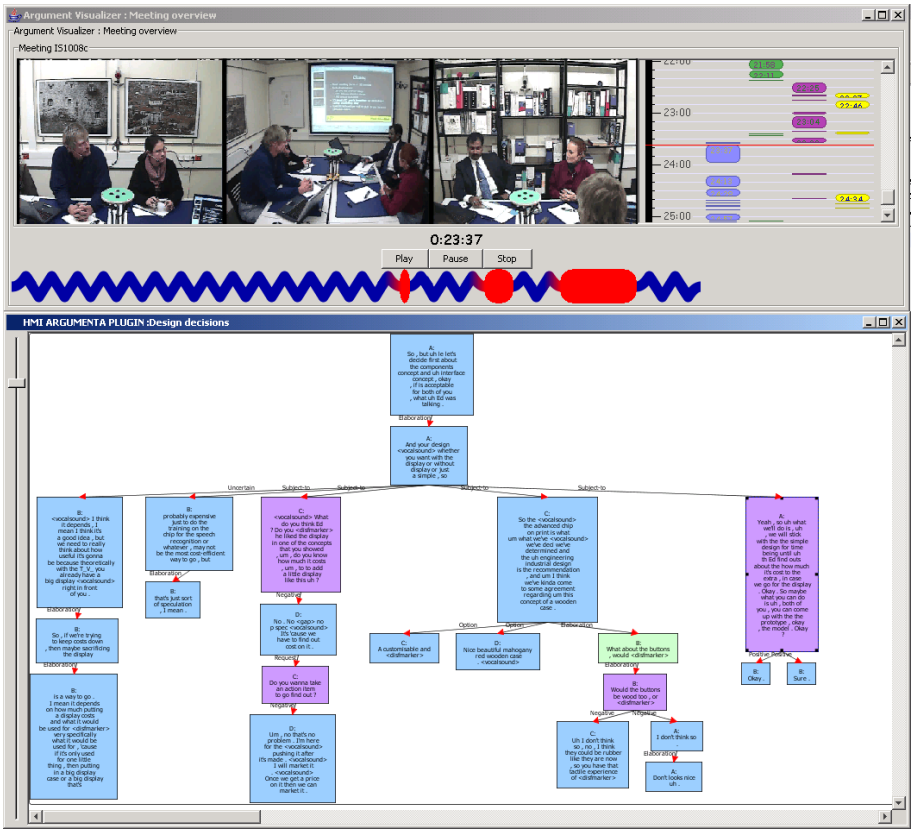


Fig. 1. Argument-Diagrams as a JFerret browser plugin

on a canvas where they can be attached to the graph via an intuitive drag and drop interface. Once attached, a popup window appears from which the relation-label can be chosen. The resulting trees can be saved in both NXT-format as well as in a specific XML format designed for this schema. The latter we used as input for our classifiers described in Section 5, whereas the former is used in for example the browser plug-in.

Three annotators were trained in several iterations. Apart from collectively developing the schema, elaborate discussions were held after a number of training sessions about when and why to pick a particular label in that particular case. The corpus of meetings that we annotated with TAS was the AMI-Corpus [Carletta et al., 2005] containing over 100 hours of meeting data. This resulted in a total of 256 annotated discussions (diagrams) including over 5000 unit labels and 5000 relation labels.

With respect to the issue of reliability one should first 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 our own, suggest that the analyst should make *plausibility judgements* rather than absolute analytical decisions, implicating more than one reasonable analysis. Furthermore there are, to our knowledge no techniques available that measure agreement between annotators for methods composed of both units as well as relations.

What we did therefore was compare the unit labels on pre-segmented discussions for four meetings (12 discussions) between two annotators. It turned out that, especially in the beginning the value of Cohen's kappa (κ) [Cohen, 1960] were rather low (0.50) as some confusion existed amongst the labels 'other' and 'statement'. When this was resolved, κ rose to a more acceptable value (0.87). Still however, the issue remained that some labels hardly occurred. The reliability issue for the relation part of the scheme is still under investigation.

We are of the opinion that for a reliable agreement measure, one needs a lot of data from a lot of annotators, which is very expensive. We used κ also in a way comparable to Steidl et al. [2005], by setting out the results of a classifier trained on (unit label) annotations of one annotator against the values provided by another annotator. (See Section 5).

4 About the Usability of Argument Diagrams

The possible applications for meetings annotated with the TAS schema are endless. They can be used for automatic summarization purposes, or aid processes aiming to find out who adhered to a specific opinion at any given moment. They can be used to see who proposed the accepted solution, or who objected to most of the discussed points. Currently we foresee these kinds of applications and are working our way towards them. See for example the work described in Rienks et al. [2006].

For an end user it is said that argument diagrams themselves provide a representation leading to quicker cognitive comprehension, deeper understanding and enhance 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 order to test the usability of the argument diagrams themselves in comparison with other representations of the same discussion, we devised the following test:

4.1 Method

Stimuli Imitating a user that wants to ask a question of a browser system, we created a list of hard to answer multiple-choice questions about the contents of six similar discussions about the design of a remote control. These questions

were shown on a screen to subjects using a newly created software package. The answers could be found in provided representations of the discussions printed on a piece of paper. Each question could be answered by selecting the answer with the mouse.

Procedure We provided the discussions in one of the following three representations: (1) a printout of the raw transcriptions of the discussion, (2) the transcription with a colored background in correspondence with the labelled unit segments of the TAS-schema and (3) a TAS-argument diagram.

Before the start of the experiment the subjects were asked to read a document describing how to read and interpret the representation of the discussions presented. Next, each subject was asked to complete as many questions as they could answer about 6 discussions. It was impossible to answer all questions within the given time frame and the same questions were asked in the same order in each of the conditions. After exactly five minutes the subjects were asked to proceed with the next discussion. Apart from giving the answer we also asked for the perceived difficulty of the question and measured the time it took to complete each question. No breaks were allowed in between the discussions. and the subjects were asked not to start reading the discussion before the first question was shown on the screen. Subjects did not receive any feedback on their judgements.

Participants A total of 30 persons (25 male and 5 female) participated in the experiment. Resulting in 10 completed experiments per condition or representation. The participants were students and employees of our department in the age range between 22 and 59.

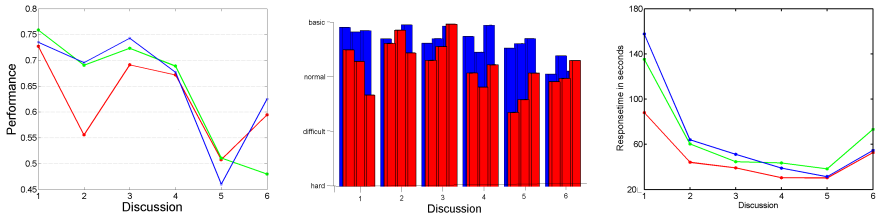
4.2 Results

Since the number of participants does not allow us to make hard inferences, our findings should therefore be regarded rather as indicative. The most important results of our experiments are shown in Figure 2. In total 848 questions were answered, from which 471 were correct.

Figure 2(a) shows the performance (percentage correctly answered questions) of the subject in each of the conditions. The blue line corresponds to the subjects using the argument diagrams, the green line for the colored transcripts, and the red line for the raw transcription. The figure shows that for the first four discussions the performance of the raw transcript is lower than for the other two conditions. Indicating that the extra information embedded in the unit labels, which is contained in both the other two conditions, could result in the observed performance increase.

When looking at the response time for the correctly answered questions, it seems that for the first two discussions the subjects need to get used to the unknown representation types. For all the discussions it appears that the raw transcription condition results in a quicker answer, although the difference for the last four discussions is much smaller than the difference for the first two discussions.

The findings for the perceived difficulty are depicted in Figure 2(b). The blue bars correspond to correctly answered questions, whereas the red bars correspond to wrongly answered questions. It appears that when the questions were answered correctly, in four out of six discussions the questions were perceived as more easy when provided with an argumentation diagram (third column), than when provided with another representation (first column = raw transcript, second column = colored transcript). It should be noted, especially for the first discussion, that not only the perceived difficulty is harder, but also the the response time was much longer.



(a) Performance (b) Difficulty (c) Response time

Fig. 2. Some results of the user experiment

Though it can be that the differences are caused by differences in the questions and the individual discussions, it appears that people need time to get used to the argument diagrams in order to reap their benefit. Once used to the method people perceive the questions as less hard and the extra information embedded in the unit labels seems to increase the performance.

5 Automatically Tagging the TAS-unit labels

Eventually we aim for a system that (a) automatically can find discussion segments, (b) can tag individual contributions with TAS-unit-labels, (c) depicts and labels the relations between the units using the TAS-relation-labels and (d) generates a visualization of the argument diagram. Here we report on our first experiments related to the automatic classification of the TAS unit labels.

5.1 Features

The features are all (except for the *lastlabel* feature) lexical features.

- **? and OR** A good indicator for an issue is a question mark. The *?-feature* gives a binary value whether a question mark is present or not. If a question mark is available, the number of times the word *or* appears is counted and used as a feature.

trigram	statement	weak statement	open issue	a/b issue	y/n issue	unknown
what do you	3	0	100	97	2	0
do you think	3	1	97	92	100	0
we have to	63	30	50	1	93	4

Table 1. Examples of trigram points

- **Length** The length (number of words) of each segment is a feature which mostly helps to make a distinction between the *statement* and *unknown* labels.
- **Last Label** Since discussions have the property of having some coherence we might expect that given the label of a segment the conditional chance of the label of the next segment might differ from the unconditional chance. Therefore the *lastlabel* feature, which is a bigram of the previous two labels, is are used.
- **N-gram points** The n-gram-point feature is used to reduce the number of features. At first, all bi-, tri- and quadri-grams are computed for all segments. Then, for each label the X most popular n-grams are selected. Each n-gram will get points according to is popularity. The most found n-gram will receive X points, the next X-1 and so on. Table 1 gives a good example of some tri-grams and their acquired points. From this, the tri-gram 'what do you' appears to be quite a good indicator for both *open issues* as well as *a/b issues*.
- **POS-ngram points** The POS n-gram-point features are quite similar to the n-gram point features. But instead of attributing points to words, points are attributed to n-grams of Part-of-Speech tags.

Perl scripts were used to extract the features *? and OR*, *Length*, and *Last Label* from our XML-format. The construction of n-grams was done using the N-gram Statistic Package (NSP) [Banerjee and Pedersen, 2003]. Using the Stanford Part-of-Speech tagger all segments were tagged to make POS-n-gramming possible [Toutanova et al., 2003].

5.2 Baseline

The corpus as it stands is unbalanced, consisting of 4245 *statements*, 199 *weak statements*, 244 *open issues*, 72 *a/b issues*, 460 *y/n issues* and 3061 *unknowns*. A baseline can be calculated by taking the share of the largest class. This would result in a baseline of 51.3% (as *statement* covers 51.3% of the occurring unit labels). Another baseline could be obtained by using the best of the features *? and OR*, *Length*, *Last Label (unigram)*. We have used the implementation of a one-rule classifier provided with the Weka toolkit [Witten and Frank, 2000] to find the feature with the smallest error. This appeared to be *Length* with a correct score of 69.1%. As can be seen in Table 3, the classifier labels each

feature	average result
? and OR	32.3
Length	33.2
Last Label	22.6

Table 2. One-rule performances on *simple* features

numberofwords	
<	4.5 − > unknown
<	5.5 − > statement
<	6.5 − > unknown
>=	6.5 − > statement

Table 3. One-rule tree

segment with the label *statement* or *unknown* and does not bother about the other labels which have a lot less instances. To see how our features would classify for a balanced corpus we have picked 50 items from each class to form a test set. More than one balanced training set was used in order to obtain more reliable results. The most simple way to obtain a baseline for each test set would again be to estimate the share of the largest class, resulting now in a baseline of 16.7%. We chose however again for an alternative baseline, by computing a baseline using the one-rule classifier. The averages of the results for each balanced set are shown in Table 2.

The feature *Length* gives the best average results, resulting in an average baseline of 33.2%. We have used two methods to train on our balanced corpus. Our first method was by using the balanced corpus itself as a test-set and using 10-fold cross validation to produce results. The second method used our total corpus minus our balanced corpus, resulting in a biased training set and an unbiased test set.

5.3 Results

We used two different Machine learning techniques to produce our results: **Weka's J48** implementation of the C4.5 decision tree algorithm [Quinlan, 1993], since this classifier gave the best results as a baseline classifer compared to seven other classifiers available in Weka and **TiMBL**, a memory-based learning system described in Daelemans et al. [2004] to compare the decision tree algorithm with a classifier using a total different machine learning algorithm. Table 4 shows a comparison of results on our balanced test set and the averaged result for four randomly picked balanced training sets. *t/t* means that the results were obtained using a different training and test set, whereas *10f* indicates that 10-fold cross validation was applied.

Unbalanced Our best result on the unbalanced corpus (88.4%) shows an improvement of 19.2% on the best baseline. This result was better than the result obtained by the TiMBL classifier using the default settings. We only computed the *10-fold* option for classifying since our unbalanced corpus does not consist of a separate training and test set. The confusion matrix produced by the J48, (Table 5) shows that improvement could be obtained by features that distinguish between utterances with the label *statement* or *unknown*.

System	Unbalanced Result	Balanced Result(Average)
Majority class baseline	16.7%	16.7%
One rule baseline *(10f)*	69.1%	30.7%
Weka J4.8 *(t/t)*	-	82.2%
Weka J4.8 *(10f)*	88.4%	81.0%
TiMBL *(t/t)*	-	59.1%
TiMBL *(10f)*	79.2%	63.4%

Table 4. Comparison of results. For the TiMBL results the *overlap metric* was used.

a	b	c	d	e	f	< −− classified as
3907	18	3	1	11	305	a = statement
33	144	0	1	3	18	b = weak statement
13	0	208	3	10	10	c = open issue
9	0	1	57	4	1	d = a/b issue
13	0	4	1	437	5	e = y/n issue
490	6	1	0	8	2556	f = unknown

Table 5. Confusion matrix of unbalanced J48-classifier

Balanced The confusion matrix resulting from applying J48 to an unbalanced training set (*t/t*) (Table 6) shows that the results are *biased to* the labels *statement* and *unknown*. Comparing this to the confusion matrix of the J48 classifier trained with a balanced training set (*10-fold*) (Table 7), the classification of the less frequent occurring labels *weak statement* and *a/b issue* show a major improvement. Our averaged best result of 82.2% shows an impressive improvement of 53.9% over our 'best' baseline.

5.4 Reliability

To get more insight into the reliability of our annotations 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 8. When both training and test sets were picked from the same annotator, we used 10-fold cross-validation. The results of Table 8 show quite high performances ranging between 75.5% and 91.9%. To focus on how useful this performance actually is, we 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 that was trained and annotator X is the test set. The resulting κ-values are shown in Table 9. It appears that annotator 3 has the highest κ-value on its own model, which implies that annotator 3 can best be imitated by a classifier. The results also

a	b	c	d	e	f	< −− classified as
46	0	0	0	0	4	a = statement
9	35	1	0	0	5	b = weak statement
3	0	43	0	3	1	c = open issue
5	0	6	32	2	5	d = a/b issue
1	0	0	0	49	0	e = y/n issue
4	0	0	0	0	46	f = unknown

Table 6. Confusion matrix of balanced J48-classifier, trained on unbalanced training set

a	b	c	d	e	f	< −− classified as
37	2	2	1	1	7	a = statement
3	41	0	3	0	3	b = weak statement
1	1	43	2	1	2	c = open issue
0	1	1	46	1	1	d = a/b issue
2	0	1	2	45	0	e = y/n issue
12	2	0	0	0	36	f = unknown

Table 7. Confusion matrix of balanced J48-classifier, trained on balanced training set

Trained / Tested on	Annotator 1	Annotator 2	Annotator 3
Annotator 1	88.6%	88.0%	76.9%
Annotator 2	86.5%	88.7%	75.5%
Annotator 3	78.0%	78.3%	91.9%

Table 8. Performance amongst annotators

indicate a rather high agreement between annotators 1 and 2, compared to the agreement between annotator 3 and the others. Analysis showed that annotator 3 made less use of the label *weak statement* and had a tendency to classify short phrases (up to 4 words) as *unknown*. Although the TAS-unit labels do consist of mutually exclusive categories the *distances* between the categories are not equal. But since we cannot, in any way, calculate the distance between the categories we computed the un-weighted κ, which potentially influenced our κ-values in a negative way. The results however indicate that a computer system can easily compete with humans on annotating TAS-unit labels.

κ	Annotator 1	Annotator 2	Annotator 3
Annotator 1	0.88	0.78	0.63
Annotator 2	-	0.89	0.62
Annotator 3	-	-	0.94

Table 9. κ values for different annotator vs. virtual annotator combinations

5.5 Discussion

The work done in this research shows much resemblance to work done for dialog-act tagging. Research in this field mostly concentrates on cues that are either manually [Hirschberg and Litman, 1993] or automatically [Reithinger and Klesen, 1997] selected. Samuel [2000] introduces an interesting baseline, called the

LIT set, in his work on DA Tagging. This set consists of 687 different cue phrases proposed in twelve papers, dissertations and books. We have constructed a feature set based on the LIT set and tested it on the unbalanced set, using the J48 classifier, which resulted in an accuracy of 71.48%, compared to a score of 86.25% when using only n-grams of words and Part-of-Speech tags. This is a reasonable result, but much worse than our best finding.

The biggest difference for our approach in comparison to earlier dialogue act classifying approaches is the use of a *compressed* feature set. Unlike other research such as carried out by Ang et al. [2005] we have not only made use of the first two words in an utterance, but of each word. But unlike Zimmermann et al. [2005] and Warnke et al. [1997] this did not result in an extremely large feature set. In addition to the compression we have made use of n-grams of POS-tags which has previously been done in research concerning the creation of backchannels in a spoken dialogue system [Cathcart et al., 2003]. The compression decreases the size of our feature vector and therefore also decreases our computing time. This of course, by itself not an advantage, unless we maintain accuracy. To compare the results of our compaction to the uncompressed set, we have constructed a feature set in the same way, though limiting it to 3000 features for complexity reasons. This resulted in an accuracy of 68.55% on the unbalanced set using the J48 classifier.

6 Conclusions

We have given an overview of the on-going work on argument diagrams of meeting discussions in Twente. We have shown with our user experiment that when used to the representation technique, it seems that finding an answer to a question is perceived less difficult, although the time required for finding the answer is somewhat larger. With respect to the learnability of the units we have shown that J48 turned out to be the the best classifier. With a performance of 88.4% on our unbalanced and an average of 82.2% on our balanced test set. The derived κ values (Section 5.4) for each of the corpus segments were in the order between 0.62 and 0.79, which is rather encouraging and indicates that a computer can compete with other annotators on labelling TAS-unit labels. We continue our research on learning machines to apply the schema to transcriptions and we will start looking into applications that use the argument diagrams as input for further (on- and off-line) enhanced user experiences.

7 ACKNOWLEDGEMENTS

This work was partly supported by the European Union 6th FWP IST Integrated Project AMI (Augmented Multi-party Interaction, FP6-506811, publication AMI-XX). We would like to thank Job Zwiers, Dennis Reidsma, Dirk Heylen, Rieks op den Akker, Betsy van Dijk, Anton Nijholt and Lynn Packwood for their Support.

Bibliography

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, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner. The ami meetings corpus. In *Proceedings of the Measuring Behavior 2005 symposium on "Annotating and measuring Meeting Behavior"*, 2005. AMI-108.

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.

W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. Timbl: Tilburg memory based learner, version 5.1, reference guide. ILK Research Group Technical Report Series 04-02, Tilburg, 2004. URL `http://ilk.uvt.nl/downloads/pub/papers/ilk0402.pdf`.

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

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.

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.

V. Pallotta, J. Niekrasz, and M. Purver. Collaborative and argumentative models of meeting discussions. In *Proceeding of CMNA-05 international workshop on Computational Models of Natural Arguments (part of IJCAI 2005)*, July 2005.

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

D. Reidsma, D.H.W. Hofs, and N. Jovanovic. A presentation of a set of new annotation tools based on the nxt api. Poster at Measuring Behaviour 2005, 2005. AMI-105.

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

R.J. Rienks and D. Heylen. Argument diagramming of meeting conversations. In A. Vinciarelli and J-M. Odobez, editors, *Multimodal Multiparty Meeting Processing, Workshop at the 7th International Conference on Multimodal Interfaces (ICMI)*, pages 85–92, Trento, Italy, October 2005.

R.J. Rienks, A. Nijholt, and P. Barthelmess. Pro-active meeting assistants : Attention please! In *Social Intelligence Design*, Osaka, Japan, March 2006.

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

S. Shum. Negotiating the construction and reconstruction of organisational memories. *Journal of Universal Computer Science*, 3(8):899–??, 1997.

S. Steidl, M. Levit, A. Batliner, E. Nöth, and H. Niemann. "of all things the measure is man" automatic classification of emotion and intra labeler consistency. In *ICASSP 2005, International Conference on Acoustics, Speech, and Signal Processing*, 2005.

K. Toutanova, D. Klein, and C. Manning. Feature-rich part-of-speech tagging with a cyclic dependency network. 2003. URL `citeseer.ist.psu.edu/620236.html`.

E. Van der Weijden. Structuring argumentation in meetings : Visualizing the argument structure. Master's thesis, University of Twente, November 2005.

T.J. van Gelder. Argument mapping with reason!able. The American Philosophical Association Newsletter on Philosophy and Computers, 2002.

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

D.N. Walton. *Argument Structure, A pragmatic Theory*. University of Toronto Press, 1996. ISBN 0-8020-0768-6.

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.

I. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. Morgan Kaufmann, 2000.

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

M. Zimmermann, Y. Liu, E. Shriberg, and A. Stolcke. A* based joint segmentation and classification of dialog acts in multiparty meetings. 2005.