A Multidimensional Approach to Utterance Segmentation and Dialogue Act Classification

Jeroen Geertzen and **Volha Petukhova** and **Harry Bunt**

Dept. of Language & Information Science,
Faculty of Humanities, Tilburg University,
P.O. Box 90153, 5000 LE Tilburg, The Netherlands,
{j.geertzen,v.petukhova,harry.bunt}@uvt.nl

Abstract

In this paper we present a multidimensional approach to utterance segmentation and automatic dialogue act classification. We show that the use of multiple dimensions in distinguishing and annotating units not only supports a more accurate analysis of human communication, but can also help to solve some notorious problems concerning the segmentation of dialogue into functional units. We introduce to use per-dimension segmentation for dialogue act taxonomies that feature multi-functionality and show that better classification results are obtained when using per-dimension segmentation than when using a single segmentation. Three machine learning techniques are applied on compared on the task of automatic classification of multiple communicative functions of utterances. The results are encouraging and indicate that communicative functions in important dimensions are well machine-learnable.

Introduction

Computer-based interpretation and generation of human dialogue is of growing relevance for today's information society. Not only is natural-language based dialogue increasingly becoming an attractive and technically feasible human-machine interface, but also the analysis of human-human interaction, for example in interviews or meetings, is important for archival and retrieval purposes, as well as for knowledge management purposes and for the study of social interaction dynamics.

Since people involved in communication constantly perceive, understand, evaluate and react to each other's intentions as encoded in statements, questions, requests, offers, and so on, a natural approach to the analysis of human dialogue behaviour is to assign meaning to dialogue units in terms of dialogue acts. The identification and automatic recognition of the dialogue acts or *communicative functions*[1] of utterances is therefore an important task for dialogue analysis and the design of applications such as computer dialogue systems.

The assignment of appropriate meanings to 'dialogue units' presupposes a way to segment a dialogue into meaningful units. This turns out to be a complex task in itself. Many previous studies in the area of the automatic dialogue act assignment were typically carried out at the level of 'utterances' or that of 'turns'. A turn can be defined as a stretch of communicative behaviour produced by one speaker, bounded by periods of inactivity of that speaker or by activity of another speaker (Allwood, 2000). While turn boundaries can be recognised relatively easily, depending on the analysis goal, a segmentation into turns is often unsatisfactory because a turn may contain several smaller meaningful parts. Utterances, on the other hand, are linguistically defined stretches of communicative behaviour that have one or multiple communicative functions. Utterances may coincide with turns but are usually smaller. The detection of utterance boundaries is a highly nontriv-

[1] In this paper, we use the terms 'dialogue act' as synonymous with 'communicative function'.

ial task. Syntactic features (e.g. part-of-speech, verb frame boundaries of finite verbs) and prosodic features (e.g. boundary tones, phrase final lengthening, silences, etc.) are often used as indicators of utterance endings (Shriberg et al., 1998; Stolcke et al., 2000; Nöth et al., 2002).

One of the problems with dialogue segmentation into utterances is that utterances may be discontinuous. Spontaneous speech in dialogue usually includes filled and unfilled pauses, self-corrections and restarts; for example, the speaker of the utterance in (1) corrects himself two times.

(1) *About half ...* ***about a quar- ... th- ...third*** *of the way down I have some hills*

Dialogue utterances may be interrupted by even more substantial segments than repairs and stallings. For example, the speaker of the utterance in (2) interrupts his Inform with a WH-Question:

(2) *Because twenty five Euros for a remote...* ***how much is that locally in pounds?*** *is too much money to buy an extra remote or a replacement remote*

Examples such as (1) and (2) show that the segmentation of dialogue into utterances that have a communicative function requires these units to be potentially discontinuous. In some cases a dialogue act may be performed by an utterance formed by parts of more than one turn. This often happens in polylogues where participants may interrupt each other or talk simultaneously. For example:

(3) *A: Well we can chat away for ... um... for five minutes or so I think at...* ***B: Mm-hmm*** *... at most*

Another case of a dialogue act that is spread over multiple turns occurs when the speaker provides complex information that is divided up into parts, in order not to overload the addressee, as in (4). The first part of the discontinuous segment that expresses S's answer also has a feedback function (making clear to U what S understood).

(4) U: Could you tell me what time there are flights to Kuala Lumpur on Monday?
S: *There are two early KLM flights, at 7.30 and at 8:25,..*
U: *Yes,...*
S: *... and a midday flight by Garoeda at 12.10,...*
U: *Yes,...*
S: *And there's late afternoon flight by Malaysian Airways at 17.55.*

The material in the three turns contributed by S together constitute the 'utterance' expressing S's answer to U's question. Examples such as these show that the units in dialogue that carry communicative functions are often very different from the traditional linguistically defined notion of an utterance. We therefore prefer to give these units a different name: *functional segment*, and we define these units as *(possibly discontinuous) stretches of communicative behaviour that have one or more communicative functions* (Bunt and Schiffrin, 2007). In many cases a functional segment corresponds to an 'utterance' as defined by certain linguistic properties, but in other cases it doesn't; and so the question arises how functional segments can be recognised. This is one of the main issues that this paper addresses.

When we want to segment a dialogue into functional segments, one complication is that of discontinuous segments, either within a turn or spread over several turns. An even greater challenge is posed by those cases where different functional segments overlap, as in the following example.

(5) U: What time is the first train to the airport on Sunday?
S: ***The first train to the airport on Sunday*** is at ...ehm... 6.17.

The first part of S's turn repeats most of the preceding question, displaying what the system has heard, and as such has a feedback function. The turn as a whole minus the part ...ehm... has the communicative function of a WH-Answer, and that part has a stalling function. So the segments corresponding to the WH-Answer and the feedback function *share* the part *The first train to the airport on Sunday*. This means that in this turn we have two functional segments starting at the same position but ending at different positions; in other words, no single segmentation of this turn exists that gives us all the relevant functional segments.

To resolve this problem adequately, we propose not to maintain a single segmentation, but to use multiple segmentations in order to allow multiple functional segments that are associated to a specific utterance to be identified more accurately. This approach is compatible to dialogue act taxonomies that address several aspects ('dimensions') of the interactive process simultaniously (e.g. DAMSL (**?**) or DIT (Bunt, 2006)), such as the task or activity that motivates the dialogue; the management of taking turns,

or timing and attention. This multidimensional view of dialogue naturally leads to the suggestion to also approach dialogue segmentation in a multidimensional way, and to segment a dialogue *per dimension* rather than in a single way. In the case of example (5), this means that S's turn is segmented in the three dimensions addressed by the functional segments in this turn:

- Dimension Task/Activity: segment the turn as consisting of the discontinuous segment *The first train to the airport on Sunday is at / 6.17*, which has a communicative function in this dimension, and the contiguous segment *...ehm...*, which does not have a function;
- Dimension Feedback: segment the turn as consisting of the contiguous segment *The first train to the airport on Sunday*, which has a function in this dimension, and the contiguous segment *is at ...ehm... 6.17*, which does not have a function;
- Dimension Time Management: segment the turn as consisting of the contiguous segment *...ehm...*, which has a communicative function in this dimension, and the discontinuous segment: *The first train to the airport on Sunday is at 6.17*, which does not have a function.

In recent work the benefits of multidimensional approaches of dialogue act annotation have been discussed and it has been argued that such approaches allow a more accurate modelling of human dialogue behaviour (Petukhova and Bunt, 2007). In this paper we report the results of two studies: one on segmentation and one on classification of dialogue acts in multiple dimensions using various machine learning techniques. In Section 1 we will outline the two series of experiments, describing the data, features, and algorithms that have been used. Section 2 and 3 reports on the experimental results on segmentation and classification, respectively. Consequently, conclusions are drawn (Section 3.1).

1 Studies outline

The first study is motivated by the question whether a different segmentation for each of the DIT dimensions (per-dimension segmentation) rather than a single segmentation for all dimensions will allow more accurate labeling of the communicative functions. In the second study we present the results of a series of experiments carried out in order to asses the automatic recognition and classification of communicative functions. For this purpose we apply machine-learning techniques. Such techniques have been already successfully used in the area of automatic dialogue processing[2]. Our approach is to train classifiers to learn communicative functions in multiple dimensions, taking functional segments as units.

1.1 Corpus data

In our experiments we used two data sets, namely, human-human dialogues in Dutch (DIAMOND corpus (Geertzen et al., 2004)) for the segmentation study and the classification study and human-human multi-party interactions in English (AMI-meetings)[3] for the classification study.

The *AMI corpus* contains manually produced orthographic transcriptions for each individual speaker, including word-level timings that have been derived using a speech recogniser in forced alignment mode. The meetings are video-recorded and each dialogue is also provided with sound files (for our analysis we used recordings made with close-talking microphones to eliminate noise). Three scenario-based[4] meetings were selected to constitute a training set of 3,676 functional segment instances.

The *DIAMOND corpus* contains human-machine and human-human Dutch dialogues that have an assistance-seeking nature. The dialogues were video-recorded in a setting where the subject could communicate with a help desk employee using an acoustic channel and ask for explanation on how to configure and operate a fax device. The dialogues were transcribed on word-chunk level and 800 utterances from the human-human subset of the corpus have been selected, for which 80% were used for training and the remaining 20% for testing.

Table 1 gives an overview of the percentage of instances for the ten most frequent occurred functional tags in both training sets.

[2] See e.g. (Clark, 2003) for an overview.

[3] Augmented Multi-party Interaction (`http://www.amiproject.org/`).

[4] Meeting participants play different roles in a fictitious design team that takes a new project from kick-off to completion

AMI data		DIAMOND data	
Tag	Percentage	Tag	Percentage
Time;STALLING	20.7	Task;INSTRUCT	14.8
Auto-FB;POS.OVERAL	18.7	Task;INFORM	7.7
Turn;Turn Keeping	7.5	Time;stall	6.5
Task;INFORM	6.8	Task;INFORM elaborate	6.3
Task;INFORM Elaborate	3.5	Auto-FB;POS.OVERAL	6.2
Task;INF.Agreement	2.5	Task;WH-Question	4.5
Task;YN-Question	2.3	Auto-FB;POS.INT	3.1
Task;SUGGEST	2.0	Task;YN-Question	2.9
Task;INFORM Justify	2.0	Task;CHECK	2.6
Task;CHECK	1.6	Task:INFORM Clarify	2.1

Table 1: *Percentage of instances for most frequent functional tags in the AMI and DIAMOND training sets.*

For the AMI training set, the majority of the dialogue units address the Task dimension (33%), followed by Auto-Feedback (21.7%), Time Management (20.3%) and Turn Management (12.5%). For the DIAMOND training set, the order for the most frequently addressed dimensions is similar with Task dimension (39.1%), followed by Auto-Feedback (19.2%), and Turn Management (16.8).

1.2 Tagset

Both data sets were annotated with the DIT^{++} tagset[5]. The DIT taxonomy distinguishes 11 dimensions (e.g. *task*, *feedback*, *turn management*,..). For each dimension, at most one communicative function can be assigned, which can either occur only in this dimension (dimension-specific[6]) or occur in all dimensions (general-purpose[7]). The tagset used in the studies contains 38 domain-specific functions and 44 general purpose functions. For both data sets the annotation is based on a single segmentation. The data set drawn from the DIAMOND corpus has additionally been segmented in each of the dimensions separately.

over the course of a day.

[5]For more information about the tagset, please visit: `http://dit.uvt.nl/`.

[6]E.g. GRABBING in the Turn Management dimension.

[7]E.g. Utterance of A in example 3, which has the communicative function of INFORM in Discourse Structuring dimension.

1.3 Features

Every communicative function is required to have some reflection in observable features of communicative behaviour, i.e. for every communicative function there are devices which a speaker can use in order to allow its successful recognition by the addressee such as linguistic cues, intonation properties, dialogue history, etc. State-of-the-art automatic dialogue understanding use all available sources to interpret a spoken utterances. Features and their selection play a very important role in supporting accurate recognition and classification of functional segments and their computational modeling may be expected to contribute to improved automatic dialogue processing. The instances in the data sets contain features related to *dialogue history*, *prosody*, and *word occurrence*.

For the AMI meetings and the DIAMOND dialogues, history consists of the functional tags of the 10 and 4 previous turns, respectively.[8]. Additionally, the functional tags of utterances of which the utterance at focus was the direct response to and the differences in start and end time with the segment in question are included as feature. For the data that has also been segmented per dimension, some segments are located inside other segments. This occurs for instance with backchannels and interruptions that do not cause turn shifting; the occurrence of these events is encoded as a feature.

Prosodic features that are included are minimum, maximum, mean, and standard deviation of *pitch* (F0 in Hz), *energy* (RMS), *voicing* (fraction of locally unvoiced frames and number of voice breaks), and *duration*. Word occurrence is represented by a bag-of-words vector using a lexicon[9] in which words are indicated as being present or absent in the segment. In total, 1.668 features are used for AMI data and 971 for DIAMOND data. For AMI data we additionally indicated the speaker (A, B, C, D) and the addressee (other participants individually or the group as a whole).

[8]We take at least twice as many tags for the AMI data since there is often more distance between related utterances in multiparty interaction than in dialogue.

[9]With a size of 1,640 entries for AMI data and 923 for DIAMOND data.

1.4 Classifiers

For many NLP tasks a wide variety of machine-learning techniques have been used with various instantiations of feature-sets and target class encodings. For applying machine-learning in dialogue processing, it is still an open issue which techniques are the most suitable for which task. We used three different types of classifiers to test their performance on our dialogue data: a probabilistic one, a rule inducer and memory-based learner.

A *Naive Bayes classifier* was used as a simple probabilistic classifier. This classifier assumes class-conditional independence, which does not always respect the characteristics of the features used. However, Naive Bayes classifiers often work quite well for complex real-world situations and are particularly suited when the dimensionality of the inputs is high. Moreover, this classifier requires relative ly little computation and can be efficiently trained.

As a rule induction algorithm, we chose *Ripper* (Cohen, 1995). The advantage of such an algorithm is that it discovers regularities in the data represented as human-readable rules.

The third classifier is IB1, which is a memory-based learner that is a successor of the k-nearest neighbour (k-NN) classifier. The algorithm stores a representation of all training examples in memory and searches for the most similar example in memory according to a similarity metric when classifying new instances, and extrapolates from k-NNs the class to the new instances. The classifier may yield more precise results, because it does not discard low-frequent phenomena from the induced knowledge model (Daelemans et al., 1999).

The results of all experiments were obtained using 10-fold cross-validation[10].As baseline we used prediction of the classes solely on the basis of one single feature, namely, the functional tag of the previous dialogue utterance (see (Lendvai et al., 2003)). For the classification score, we use accuracy (percentage of true negatives plus true positives from all instances).

[10]In order to reduce the effect of imbalances in the data, it is partitioned ten times. Each time a different 10% of the data is used as test set and the remaining 90% as training set. The procedure was repeated ten time so that in the end, every instance has been used exactly once for testing (Witten and Frank, 2000). The cross-validation was stratified, i.e. the 10 folds contained approximately the same proportions of instances with relevant tags as the entire dataset.

2 Multidimensional dialogue act segmentation

Any segmentation of dialogue (or multi-party interaction) into meaningful units, such as functional segments, is motivated by the meaning that is conveyed. As a result, the segmentation strongly depends on the definition of the dialogue acts in the taxonomy that is used. The multidimensional tagset used in this paper allows to address several aspects of communicative behaviour for a single functional segment. However, the functions of a segment do not necessarily address the same span in the communicative channels. Hence it could be argued that per-dimension segmentation should allow for a more accurate identification of spans associated to specific communicative functions. Assuming this to be the case, it would follow that classification of communicative functions based on dimension-specific segments should be more successful than classification based on a single segmentation.

For testing this, we use *Ripper*, the classifier that provides the best classification results that we found. Running Ripper with default parameters for both the single and per-dimension segmentation results in the scores presented in Table 2:

Dimension	uniform	specific
Task	67.2	68.7
Auto Feedback	82.1	84.6
Allo Feedback	98.4	99.6
Turn Management	88.3	90.0
Time Management	70.2	73.1
Contact Management	97.1	97.1
Topic Management	53.1	53.1
Own Com. Management	84.6	85.7
Partner Com. Management	67.3	67.3
Dialogue Struct. Management	74.0	74.0
Social Obl. Management	95.4	95.4

Table 2: Accuracy scores for communicative function labeling grouped per dimension on single and per-dimension segmentation.

From the results in Table 2 we can observe that for the most important dimensions, per-dimension segmentation results in better classification performance. The functions related to the dimensions Task, Auto Feedback, Turn Management, and Time

Management are particularly favoured by a per-dimension segmentation.

Although not all dimensions benefit significantly from per-dimension segmentation, it seems clear that multidimensional segmentation helps to classify communicative functions more accurately. Two interesting directions in which this study can be extended are, first, to (manually) segment more dialogue data both for single and per-dimension segmentation and to see the effect of the larger data set on the classification performance with both segmentations. Second, it would be interesting to repeat a similar experiment on corpus material which allows to consider more modalities than only speech audio, such as the AMI data used.

3 Dialogue Act Classification in Multiple Dimensions

Since a functional segment is often multi-functional, it is interesting to not only identify communicative function per dimension separately and the functional tag as described above, but also to test whether and to what extent is it possible to learn multiple functional tags which is practically the combination of pairs as described above (e.g. Time:STALLING;Turn:KEEP).

We carried out a set of experiments studying the performance of the three classifiers described in Section 1 on the following tasks:

- addressed dimension or multiple dimensions, e.g. Task, Auto-Feedback, Turn Management, etc.;
- communicative function per dimension in isolation;
- functional tag(-s) (either $\langle D, GP \rangle$ or $\langle D, DS \rangle$, where D stands for dimension, GP - general purpose function and DS - dimension specific function);

3.1 Experimental results

Table 3 gives an overview of success scores expressed as the percentage correctly predicted classes in all training experiments in comparison to baseline scores.

As for the prediction of dimension addressed by a functional segment all algorithms outperform the baseline by a broad margin. Ripper clearly outperforms the other two learners. As was to be expected for prediction of the Task dimension, the bag-of-words feature representing word occurrence in the segment was an important feature. For example, significant to identify INFORM JUSTIFY was the presence of *'because'* in a segment, for INFORM EXEMPLIFY the occurrence of *'like'* or *'for example'*, or *'maybe'* or *'might'* for SUGGESTIONS. Also the duration of the segment was usually longer than, for example, segments which addresses the Time or Turn Management dimensions. As for questions along with word occurrence (e.g. occurrence of wh-words in WH-Questions, and *'or'* for Alternative Questions) the prosody, features like standard deviation in pitch, was the essential source of key-features. For the segments which are identified as having Information-Providing functions, important features were detected in the dialogue history, e.g. CONFIRM about the Task was a response to the previous CHECK question about the Task. The segments addressing the Auto-Feedback dimension were classified successfully on the basis of their word occurrence and dialogue history. The occurrence of words like *alright, right, okay, uh-huh* are important clues for their recognition. As for Turn and Time Management, the duration of the segment was a key-feature, because as a rule these segments are shorter than others. Also, these utterances were pronounced softer (e.g. $<$49dB) and are less voiced (e.g. about 47% of unvoiced frames). They usually occur inside 'larger' segments, mostly in the beginning or in the middle. If they appear in clause-initial position they normally have Turn initial functions (TAKE, ACCEPT, GRAB) and the Time Manage-

Classification task	BL	NBayes	Ripper	IB1
Dimension tag	38.0	69.5	**72.8**	50.4
Task management	66.8	71.2	**72.3**	53.6
Auto-Feedback	77.9	86	**89.7**	85.9
Turn initial	93.2	92.9	93.2	88
Turn closing	58.9	85.1	**91.1**	69.6
Time management	69.7	99.2	**99.4**	99.5
Own Communication Management	89.6	90	**94.1**	85.6
Functional tag	25.7	48.0	**50.2**	38.9

Table 3: *Overview of accuracy on the baseline (BL) and the classifiers on all classification tasks*

ment function of STALLING; if they occur in the middle of the 'main' segment they are used to signal that the speaker has some difficulties to complete his/her utterance, needs some time and wants to keep the turn (see examples 3 and 5). Of course, words like *'um', 'well'* but also lengthening the words indicate the speaker's hesitation and/or difficulties in utterance completion. Segments having communicative functions in the dimension of Discourse Structuring often have linguistic cues like *'meeting', 'finish', 'wrap up'*, etc. As for RETRACTs (dimension of Own Communication Management), their relation to what is actually retracted ('reply_to' feature), but also the energy (i.e. they are pronounced harder than the retracted 'reparandum'; $>$55dB) were important attributes to be successfully classified[11].

Table 3 gives an overview of the performance of the tested classifiers on communicative functions per dimension. Ripper again outperforms Naive Bayes and IB1. The scores are the same (e.g. Turn initial functions) or higher then those of the baseline.

For some of the dimensions distinguished in DIT we do not present the results in the Table 3 since the segments which were tagged as having communicative functions in the dimensions of Allo-feedback, Contact management, Topic management, Discourse structuring, Partner Communication management and Social Obligation management are rare in the AMI training data. The instances from these dimensions were almost perfectly classified by all classifiers, reaching a success score higher than 99%, but not better than those of the baseline.

Looking further at the results we can observe that functional tag(-s) labels were difficult to classify. They eventually reach a success score of 50.2% (baseline: 25.7%). These scores should be evaluated in the light of the relatively high degree of granularity of these tags (97 unique functional tags and 132 unique combinations of functional tags) and relatively lower frequency of each of those in the training sets. We have however reason to expect that the more examples are added to the training set the higher accuracy could be reached. We aim to prove this in the future by working with larger data sets.

[11]Selection of the RIPPER induced rules with examples is presented in Appendix A of this paper

Conclusions

In this paper a multidimensional approach to utterance segmentation and automatic dialogue act classification has been presented in which some problematic issues with the segmentation of dialogue into functional units are addressed.

Whereas it is common practice to assign dialogue acts to a single segmentation or a segmentation per turn, we conclude that for dialogue act taxonomies that allow assignment of multiple functions to dialogue units we can describe human communication more accurately by using multidimensional segmentation instead.

We have shown that machine learning techniques could be profitably used on a complex task such as automatic recognition of the multiple communicative functions of dialogue segments. All three classifiers that have been tested performed well on all classification tasks. For the majority of tasks the scores we obtained scores that are significantly higher than those of the baseline. However, the datasets that we used were not very rich with respect to all the communicative functions distinguished in the various dimensions: some classes were underrepresented.

For future work, we intend to improve classification results and get a fair indication of the classification performance of general purpose functions in other dimensions than the Task and Feedback dimensions by extending the data sets with a sufficient number of instances for each class. Furthermore, we plan to increase the size of our dataset and to consider multi-modal interactions in order to study the effect of the bigger and richer data set on the classification performance when comparing per-dimension and single dimension segmentation.

References

Jens Allwood. 2000. An activity-based approach to pragmatics. In Harry Bunt and William Black, editors, *Abduction, Belief and Context in Dialogue; Studies in Computational Pragmatics*, pages 47–80. John Benjamins, Amsterdam, The Netherlands.

Harry Bunt and Amanda Schiffrin. 2007. Defining interoperable concepts for dialogue act annotation. In *Proceedings of the Seventh International Workshop on Computational Semantics (IWCS)*, pages 16–27.

Harry Bunt. 2006. Dimensions in dialogue annotation. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*.

Alexander Clark. 2003. Machine learning approaches to shallow discourse parsing: A literature review. IM2.MDM Project Deliverable, March.

William W. Cohen. 1995. Fast effective rule induction. In *Proceedings of the 12th International Conference on Machine Learning (ICML'95)*, pages 115–123.

Mark G. Core and James F. Allen. 1997. Coding dialogues with the DAMSL annotation scheme. In *Working Notes: AAAI Fall Symposium on Communicative Action in Humans and Machines*, pages 28–35.

Walter Daelemans, Antal van den Bosch, and Jakub Zavrel. 1999. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34(1/3):11–43.

Jeroen Geertzen, Yann Girard, and Roser Morante. 2004. The diamond project. Poster at the 8th Workshop on the Semantics and Pragmatics of Dialogue (CATALOG 2004)

Piroska Lendvai, Antal van den Bosch, and Emiel Krahmer. 2003. 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.

Elmar Nöth, Anton Batliner, Volker Warnke, Johannes-Peter Haas, Manuela Boros, Jan Buckow, Richard Huber, Florian Gallwitz, Matthias Nutt, and Heinrich Niemann. 2002. On the use of prosody in automatic dialogue understanding. *Speech Communication*, 36(1-2):45–62.

Volha V. Petukhova and Harry Bunt. 2007. A multidimensional approach to multimodal dialogue act annotation. In *Proceedings of the Seventh International Workshop on Computational Semantics (IWCS)*, pages 142–153.

Elizabeth Shriberg, Rebecca Bates, Andreas Stolcke, Paul Taylor, Daniel Jurafsky, Klaus Ries, Noah Coccaro, Rachel Martin, Marie Meteer, and Carol Van Ess-Dykema. 1998. Can prosody aid the automatic classification of dialog acts in conversational speech? *Language and Speech (Special Issue on Prosody and Conversation)*, 41(3-4):439–487.

Andreas Stolcke, Klaus Ries, Noah Coccaro, Elizabeth Shriberg, Rebecca Bates, Daniel Jurafsky, Paul Taylor, Rachel Martin, Carol Van Ess-Dykema, and Marie Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373.

Ian H. Witten and Eibe Frank. 2000. *Data mining: practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann Publishers, San Francisco:CA, USA.

Appendix A: Selected RIPPER rules illustrated with examples from the corpus

The structure of a rule is: if (feature = x) and (feature= x, etc.) $\Longrightarrow$ class (*n/m*), where x is a nominal feature value, an element of a set feature, or a range of a numeric feature; *n* indicates the number of instances a rule covers and *m* the number of false predictions. We illustrate the induced rules with some interesting examples from the training set.

Task Management:

(it = p) and (wouldnt = p) $\Longrightarrow$ da=task:check (5.0/1.0)
(right = p) and (max.pitch <= 203.87) $\Longrightarrow$ da=task:check (8.0/2.0)
Example:
(1052:88-1057:12) D: We were given sort of an example of a coffee machine or something, right? (dimension: Task, GP:CHECK; FT: *task:check*)

(reply_to = task;ynq) $\Longrightarrow$ da=task:yna (60.0/22.0)
(reply_to = task;ynq;t_give) $\Longrightarrow$ da=task:yna (2.0/0.0)
(reply_to = task;ynq;t_grab) $\Longrightarrow$ da=task:yna (2.0/0.0)
(reply_to = task;ynq;t_release) $\Longrightarrow$ da=task:yna (3.0/1.0)
Example:
(1407:56-1413:72) B: Do you think maybe we need like further advances in that kind of area until it's worthwhile incorporating it though (dimension:Task; GP: YN-QUESTION; FT: *task:ynq*)
(1412:96-1415:6) C: I , think , it'd , probably , quite , expensive , to , put , in (dimension:Task; GP: YN-ANSWER; FT: *task:yna*)

(yeah = p) and (dss_reply <= -3.920044) and (duration >= 0.56) and (min.pitch >= 95.007) $\Longrightarrow$ da=task:inf.agree (27.0/8.0)
(yeah = p) and (fraction:voiced/unvoiced >= 0.36634) and (dss_reply ¡= -0.52002) and (fraction:voiced/unvoiced <= 0.46875) $\Longrightarrow$ da=task:inf.agree (8.0/1.0)
(yeah = p) and (energy >= 56.862651) and (mean.pitch <= 144.971) $\Longrightarrow$ da=task:inf.agree (9.0/2.0)
(dss_reply <= -0.359985) and (sure = p) and (max.pitch <= 187.065) $\Longrightarrow$ da=task:inf.agree (8.0/0.0)
(yeah = p) and (U3 = turn:t_keep;time:stal) $\Longrightarrow$ da=task:inf.agree (14.0/6.0)
Example:
(1277:88-1286:28) D: but people who are about forty-ish and above now would not be so dependent and reliant on a computer or mobile phone (dimension:Task; GP:INFORM; FT:*task;inf*)
(1284:32-1286:16) D: Yeah, sure (dimension: Task; GP:INFORM AGREEMENT; FT: *task:inf.agree*)

(problem = p) $\Longrightarrow$ da=task:inf.warn (7.0/3.0)
(because = p) $\Longrightarrow$ da=task:inf.just (33.0/7.0)
(cause = p) $\Longrightarrow$ da=task:inf.just (26.0/9.0)
(dss_reply <= -1.52002) and (voice_breaks >= 4) and (energy >= 54.435098) and (mean.pitch <= 173.572) $\Longrightarrow$ da=task:inf.ela (51.0/21.0)
Example:
(1396:84-1403:76) C: One problem with speech recognition is the technology that was in that one wasn't particularly amazing (dimension: Task; GP: INFORM WARNING; FT: *task:inf.warn*)

(maybe = p) and (dss_reply >= 0) $\Longrightarrow$ da=task:suggest (38.0/11.0)
(duration >= 2.12) and (reply_to = _) and (might = p) $\Longrightarrow$ da=task:suggest (12.0/4.0)
Example:
(1694:6-1703:48) B: It might be a good idea just to restrict our creative influence on this and not worry so much about how we transmit it (dimension:Task; GP: SUGGESTION; FT:*task;suggest*)
(1704:4-1708:44) B: because I mean it tried and tested intra-red (dimension:Task; GP: INFORM JUSTIFY; FT:*task:inf.just*)

Auto-Feedback:
(dss_reply <= -0.039978) and (break <= 1) $\Longrightarrow$ da=au_f:au_f_p_ex (168.0/24.0)
(dss_reply <= -0.039917) and (duration <= 1.08) and (okay = p) $\Longrightarrow$ da=au_f:au_f_p_ex (84.0/8.0)
(dss_reply <= -0.039978) and (break <= 1) and (mmhmm = p) $\Longrightarrow$ da=au_f:au_f_p_ex (34.0/1.0)
(dss_reply <= -0.039978) and (break <= 3) and (voclaugh = p) $\Longrightarrow$ da=au_f:au_f_p_ex (25.0/2.0)
(okay = p) and (energy <= 56.617891) and (duration >= 1.16) $\Longrightarrow$ da=au_f:au_f_p_ex (21.0/4.0)
Example:
(1728:36-1729:88) A: Then you need to send the signal out (dimension: Task; GP:INFORM; FT:*task:inf*)
(1729:8-1730:2) B: Mmhmm (dimension: Auto-Feedback; DS: POS.EXECUTION; FT: *au_f:au_f_p_ex*)

(within = turn:t_keep;time:stal) and (duration <= 0.44) $\Longrightarrow$ da=au_f:au_f_p_ex;turn:t_give (83.0/11.0)
(within = turn:t_keep;time:stal) and (energy <= 50.235299) $\Longrightarrow$ da=au_f:au_f_p_ex;turn:t_give (9.0/2.0)
Example:
(1285:32-1292:36) B: you're gonna have audio which is gonna be like you know
B: um and (dimension:Time/Turn; DS: STALLING/T_KEEPING; FT: *turn:t_keep;time:stal*)
(1289:44-1290:08)A: mmhm (dimension: Auto-Feedback/Turn; DS: POS.EXECUTION/T_GIVING; FT: *au_f:au_f_p_ex;turn:t_give*)
B: your bass settings and actual volume hi

Turn Management:
(um = p) and (dss_reply <= -1.199997) $\Longrightarrow$ da=turn:t_acc;t_keep;time:stal (13.0/6.0)
(well = p) and (dss_within <= -0.159912) and (duration <= 0.72) $\Longrightarrow$ da=turn:t_grab;t_keep (9.0/3.0)
(um = p) and (dse_within >= 0.040039) and (dse_within <= 1.040039) and (min.pitch >= 107.875) $\Longrightarrow$ da=turn:t_grab;t_keep;time:stal (18.0/4.0)
(well = p) and (dss_within <= -1.119995) $\Longrightarrow$ da=turn:t_grab;t_keep;time:stal (6.0/2.0)
(um = p) and (dse_within <= 0) and (energy <= 49.86226) and (mean.pitch >= 114.669) $\Longrightarrow$ da=turn:t_take;t_keep;time:stal (21.0/10.0)
Examples:
(819:08-821:88) D: Well like um (dimension: Turn/Time; DS:T_GRABBING/STALLING; FT: *turn:t_grab;t_keep;time:stal*)
D: maybe what we could use is a sort of like a example of a successful other piece technology is palm pilots

Topic Management:
(back = p) and (go = p) $\Longrightarrow$ da=topic:suggest (5.0/2.0)
Example:
(1587:16-1591:72) A: I guess we should maybe go back to what the functions are (dimension: Topic Management; GP: SUGGESTION; FT:*topic:suggest*)

Discourse Structuring:
(end = p) and (min.pitch >= 175.915) $\Longrightarrow$ da=ds:inf (2.0/0.0) (wrap = p) and (U3 = au_f:au_f_p_ex) $\Longrightarrow$ da=ds:inf (2.0/0.0)
Examples:

(978:6- 981:68) D: so just to wrap up the next meeting's gonna be in thirty minutes (dimension: Discourse Structuring; GP:INFORM; FT: *ds:inf*)
(1036:44-1037:68) B: And that's the end of the meeting (dimension: Discourse Structuring; GP:INFORM; FT: *ds:inf*)

Contact Management:
ready = p) $\Longrightarrow$ da=contact:check (2.0/0.0)
Example:
(34:06-35:56) B: All ready to go? (dimension: Contact Management; GP: Check; FT: *contact:check*)

Own Communication Management:
(oh = p) $\Longrightarrow$ da=ocm:error (7.0/3.0)
(reply_to = time;t_keep;stal) and (duration >= 0.36) and (U5 = turn:t_keep;time:stal) $\Longrightarrow$ da=turn:t_keep;ocm:retract (12.0/5.0)
(reply_to = time;t_keep;stal) and (energy >= 55.581619)$\Longrightarrow$ da=turn:t_keep;ocm:retract (185.0/17.0)
(dse_within >= 0.679993) and (duration <= 0.24) and (min.pitch >= 107.013) and (max.pitch <= 155.745) and (mean.pitch >= 122.459) $\Longrightarrow$ da=turn:t_keep;ocm:retract (17.0/4.0)
Example:
(96:32-96:68) B: Oh (dimension: Own Communication Management; DS: Error; FT: *ocm:error*
B: I have to record who's here actually

Social Obligation Management:
(thanks = p) $\Longrightarrow$ da=som:thanking (2.0/0.0)
(reply_to = som;ini_selfintro) $\Longrightarrow$ da=som:react_selfintro (4.0/1.0)
Examples:
(72:8-74:44) B: I'm Laura and I'm the project manager (dimension: Social Obligation Management; DS: INITIATE SELF-INTRODUCTION; FT:*som;ini_selfintro*)
(77:44-77:76) A: I'm David and I'm supposed to be an industrial designer(dimension: Social Obligation Management; DS: REACT SELF-INTRODUCTION; FT:*som;react_selfintro*)