A Multidimensional Approach to Utterance Segmentation and Dialogue Act Classification

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

Dept. of Communication & Information Sciences,
Tilburg University, The Netherlands,
{j.geertzen,v.petukhova,h.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 the use of per-dimension segmentation for dialogue act taxonomies that feature multi-functionality and show that better classification results are obtained when using a separate segmentation for each dimension than when using one segmentation that fits all dimensions. Three machine learning techniques are applied and 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 easy machine-learnable.

1 Introduction

Computer-based interpretation and generation of human dialogue is of growing relevance for today's information society. As natural language based dialogue is increasingly becoming an attractive and technically feasible human-machine interface, so the analysis of human-human interaction (for example in interviews or meetings) is becoming 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, for some analysis 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.

[1]In this paper, we use the terms 'dialogue act' and 'communicative function' synonymously.

The detection of utterance boundaries is a highly nontrivial 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 is providing complex information and divides it up into parts in order not to overload the addressee, as is shown 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, that of *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 does not; 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 as we have already discussed. An even greater challenge is posed by those cases where different functional segments overlap, as in the example shown in 5.

(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 ones; 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 with dialogue act taxonomies that address several aspects ('dimensions') of the interactive process simultaneously (e.g. DAMSL (Core and Allen, 1997) 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 of approaching dialogue segmentation in a similarly multidimensional way, and to allow the segmentation of a dialogue *per dimension* rather than in one fixed 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;
- 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;
- 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.

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 2 we will outline the two series of experiments describing the data, features, and algorithms that have been used. Section 3 and 4 report on the experimental results on segmentation and classification, respectively. Consequently, conclusions are drawn in Section 5.

2 Studies outline

The first study is motivated by the question of 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 labelling of the communicative functions. In the second study we present the results of a series of experiments carried out in order to assess the automatic recognition and classification of communicative functions. For this purpose we apply machine-learning techniques. Such techniques have already successfully been 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.

2.1 Corpus data

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

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 explanations on how to configure and operate a fax machine. The dialogues were orthographically transcribed and 952 utterances representing 1,408 functional segments from the human-human subset of the corpus have been selected.

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 short range microphones to eliminate noise). Three scenario-based[4] meetings were selected to constitute a training set of 3,676 functional segment instances.

[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 over the course of a day.

Table 1 gives percentages of occurrence of the ten most frequently observed tags in both training sets.

AMI data		DIAMOND data	
Tag	Perc.	Tag	Perc.
Time;STALLING	20.7	Task;INSTRUCT	14.8
Auto-FB;POS.OVERALL	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.OVERALL	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 tags in the AMI and DIAMOND training sets.

For the DIAMOND training set, the order for the most frequently addressed dimensions is similar with Task dimension (45.6%), followed by Auto-Feedback (19.2%), and Turn Management (16.8%). 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%).

2.2 Tagset

Both data sets were annotated with the DIT^{++} tagset[5]. The DIT taxonomy distinguishes 11 dimensions, addressing information about: the domain or task (*Task*), feedback on communicative behaviour of the speaker (*Auto-feedback*) or other interlocutors (*Allo-feedback*), managing difficulties in the speaker's contributions (*Own-Communication Management*) or those of other interlocutors (*Partner Communication Management*), the speaker's need for time to continue the dialogue (*Time Management*), establishing and maintaining contact (*Contact Management*), about who should have the next turn (*Turn Management*), the way the speaker is planning to structure the dialogue (*Dialogue Structuring*), introducing, changing or closing the topic (*Topic Management*), and the information motivated by social conventions (*Social Obligations Management*).

For each dimension, at most one communicative function can be assigned, which can either occur in this dimension alone (the function is *dimension specific*) or occur in all dimensions (the function is *general purpose*). For example, the utterance in 1 has a dimension-specific function SELF CORRECTION assigned to it that can only be assigned in the *Own Communication Management* dimension. Utterance A in example 3 has the communicative function of INFORM in the *Dialogue Structuring* dimension. Being a *general purpose* function, INFORM could possibly also be assigned to any other dimension (such as e.g. *Task*).

The tagset used in the studies contains 38 domain-specific functions and 44 general purpose functions. As a result of difference in function type, a tag consists either of a pair of the addressed dimension (D) and general purpose function (GP) or the addressed dimension and dimension specific function (DS). Some functional segments can address several dimensions simultaneously. For example, utterances like *uhm.., ehm..* have the communicative function of STALLING in the dimension *Time Management*, but also have the TURN KEEPING function in the *Turn Management* dimension. These utterances typically have two $\langle D, DS\rangle$ tags assigned: $\langle TimeM, STALLING\rangle$ and $\langle TurnM, KEEPING\rangle$.

For both data sets the annotation is first carried out on a single segmentation and then additionally on dialogue segmented in each of the dimensions separately.

2.3 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 uses all available sources to interpret a spoken utterance. Features and their selection play a very important role in supporting accurate recognition and classification of functional segments and their computational modelling may be expected to contribute to improved automatic dia-

[5] For more information about the tagset and the dimensions that are identified, please visit:
http://dit.uvt.nl/

logue processing. The features included in the data sets are those relating to *dialogue history*, *prosody*, and *word occurrence*.

For the AMI meetings and the DIAMOND dialogues, history consists of the tags of the 10 and 4 previous turns, respectively[6]. Additionally, the tags of utterances to which the utterance in focus was a direct response to, as well as timing, are included as features. For the data which is 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[7] indicating the presence or absence of words in the segment. In total, 1,668 features are used for AMI data and 947 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).

2.4 Classifiers

A wide variety of machine-learning techniques has been used for NLP tasks with various instantiations of feature-sets and target class encodings, and for 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.

For a probabilistic classifier we used *Naive Bayes*. 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 suitable for situations in which the dimensionality of the input is high. Moreover, this classifier requires relatively little computation and can be efficiently trained.

For rule induction algorithm, we chose *Ripper* (Cohen, 1995). The advantage of such an algorithm is that the regularities discovered in the data are 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 first stores a representation of all training examples in memory. When classifying new instances, it searches for the k most similar examples (nearest neighbours) in memory according to a similarity metric, and extrapolates the target class from this set to the new instances. The algorithm may yield more precise results given sufficient training data, because it does not abstract away low-frequent phenomena during the learning (Daelemans et al., 1999).

The results of all experiments were obtained using 10-fold cross-validation[8]. When setting a baseline it is common practice to predict the majority class tag, but for our data sets such a baseline is not very useful because of the relative low frequencies of the tags in most dimensions. Instead, we use a baseline that is based on a single feature, namely, the tag of the previous dialogue utterance (see (Lendvai et al., 2003)).

3 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 several aspects of communicative behaviour for a single functional segment to be addressed. However, the functions of a segment do not necessarily address the same span in the communicative channels. Hence it could be argued that separate segmentation for each dimension should al-

[6]We use more preceding tags for the AMI data than for the DIAMOND data since there is often more distance between related utterances in multi-party interaction than in dialogue.

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

[8]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 is repeated ten times so that in the end, every instance has been used exactly once for testing (Witten and Frank, 2000) and the scores are averaged. The cross-validation was stratified, i.e. the 10 folds contained approximately the same proportions of instances with relevant tags as in the entire dataset.

low for a more accurate identification of spans associated to specific communicative functions. When we assume that this is the case, it would follow that classification of communicative functions based on per-dimension segments should be more successful than classification based on a single segmentation for all dimensions.

For testing the above-mentioned hypothesis, *Ripper* —the classifier that provides the highest accuracy scores in our experiments— was used on the DIAMOND dialogues annotated with the DIT^{++} tagset. Two classification tasks on exactly the same dialogues with exactly the same kind of features and annotated communicative functions were performed. The only difference being that in one task *one segmentation that fits all dimensions (OSFAD)* was used, whereas in the other task *per-dimension segmentation (PDS)* was used. Because DIT allows the assignment of at most one function in a specific dimension, a segment in the PDS task has one tag whereas a segment in the OSFAD setting might have a combination of tags[9]. Running *Ripper* (with default parameters) for both tasks resulted in the scores presented in Table 2:

Dimension	OSFAD	PDS	
Task	66.1	72.8	$*$
Auto Feedback	80.4	86.3	$*$
Allo Feedback	98.4	99.6	
Turn M.	88.3	90.0	
Time M.	72.6	82.1	$*$
Contact M.	97.3	97.3	
Topic M.	55.2	55.2	
Own Communication M.	85.9	87.1	
Partner Communication M.	64.5	64.5	
Dialogue Structuring	74.3	74.3	
Social Obligations M.	93.2	93.3	

Table 2: Accuracy scores for communicative functions with one segmentation that fits all dimensions (OSFAD) and per-dimension segmentation (PDS).
$*$ significant at $p < .05$, one-tailed z-test.

From the results in Table 2 we can observe that for most important dimensions, PDS results in better classification performance: the functions related to the dimensions *Task*, *Auto Feedback*, and *Time Management* show significant improvement. For some dimensions, classification does not take advantage of PDS, mainly because of two reasons: in the dataset some dimensions are rarely addressed (e.g. *Partner Communication Management*) and some dimensions are addressed without any other dimension being addressed around the same time (e.g. *Contact Management*). These observations are motivated by the kinds and characteristics of interaction and in some extend by the limited size of the dataset.

Although not all dimensions benefit significantly, it is clear that multidimensional segmentation helps to classify communicative functions more accurately. However, it should be noted that the gain of more accurately identified functions comes at the cost of a slightly more complex segmentation procedure.

[9]In our data, at most four functions occurred simultaneously.

4 Dialogue Act Classification in Multiple Dimensions

Since a segment is often multi-functional, it is not only interesting to identify the dimension, the communicative function, and the tag separately, but also to test whether or not and to what extent it is possible to learn the combination of tags (e.g. $\langle TimeM, STALLING\rangle$, $\langle TurnM, KEEP\rangle$).

We carried out a set of experiments studying the performance of the three classifiers described in Section 2 on the following classification tasks:

- each addressed dimension separately or multiple addressed dimensions in combination, e.g. a single dimension like *Task*, *Auto-Feedback*, *Turn Management*, or a combination like *Turn Management* and *Time Management*;
- communicative function per dimension in isolation, e.g. INFORM, CORRECTION, WH-QUESTION, etc. in the *Auto-Feedback* dimension;
- tag or combination of tags, e.g. either $\langle D, GP\rangle$ or $\langle D, DS\rangle$, or $\langle D, GP\rangle$,$\langle D, DS\rangle$ or $\langle D, DS\rangle$,$\langle D, DS\rangle$.

4.1 Experimental results

Table 3 gives an overview of classification scores expressed as the percentage of correctly predicted classes in all training experiments.

For the prediction of a dimension addressed by a functional segment (upper data row in the table)

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	86.0	**89.7**	85.9
Turn initial	93.2	92.9	93.2	88.0
Turn closing	58.9	85.1	**91.1**	69.6
Time management	69.7	99.2	**99.4**	99.5
OCM	89.6	90.0	**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

all algorithms outperform the baseline by a broad margin. Ripper clearly outperforms the other two learners. The middle part of the table 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. with turn initial functions) or higher then those of the baseline. Some of the dimensions distinguished in DIT are not included in Table 3 since the segments which were tagged as having communicative functions in the dimensions *Allo-feedback*, *Contact management*, *Topic management*, *Dialogue 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 an accuracy higher than 99%, but not better than those of the baseline.

In Appendix A of this paper we present a selection of the RIPPER induced rules illustrated with examples from the corpus. As was to be expected, for the prediction of the *Task* dimension, the bag-of-words feature representing word occurrence in the segment was important. For example, the presence of 'because' in a segment was a good indicator for identifying INFORM JUSTIFY; the occurrence of 'like', or 'for example', or 'maybe' and 'might' for SUGGESTION. Also the duration of the segment was usually longer than for example segments which addressed the *Time* or *Turn Management* dimensions. For the prediction of questions, word occurrence (e.g. occurrence of wh-words in WH-Questions, and 'or' for Alternative Questions) and prosodic features like standard deviation in pitch were essential. 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 a 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 the dimensions *Turn* and *Time Management*, the duration of the segment was a key feature, because the duration of these segments tends to be shorter than that of others. Moreover, these utterances were pronounced more softly (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 usually have turn initial functions (TAKE, ACCEPT, GRAB) and the function STALLING in the *Time Management* dimension; if they occur in the middle of the 'main' segment they are used to signal that the speaker has some difficulties in completing his/her utterance, needs some time and wants to keep the turn (see examples 3 and 5). Of course, usage of words like 'um', 'well', but also lengthening the words indicates the speaker's hesitation and/or difficulties in utterance completion.

Segments having communicative functions in the dimension *Dialogue Structuring* often have linguistic cues like 'meeting', 'finish', 'wrap up', etc. Important cues for RETRACTs (in the dimension *Own Communication Management*) are their relation to what is actually retracted ('reply_to' feature), and the energy with which they are spoken (i.e. they are pronounced louder than the retracted 'reparandum', i.e. $>$55dB).

Looking further at the results we can observe that tag labels were difficult to classify (see bottom data row of the table). They eventually reach an accuracy 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 tags and 132 unique combinations of tags) and relatively lower frequency of each of those in the training sets. We have however reason to expect that by increasing the size of the training set higher accuracy could be reached.

5 Conclusions and future work

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, 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 per-dimension segmentation instead.

We have shown that machine learning techniques can be profitably used on a complex task such as the automatic recognition of 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 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 extended the studies into two directions. First, we plan to increase the size of our dataset to obtain a sufficient number of instances for each class by manually segmenting and annotating more dialogue data with both segmentations. This would allow us to get a fair indication of the classification performance of general purpose functions in dimensions other than *Task* and *Feedback*. Furthermore, we plan to consider multi-party interactions (the AMI sessions for instance) and use other modalities besides speech audio in comparing both segmentations. We expect that for such data, dialogue act classification may benefit more from using per-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 corpus examples

The structure of a rule is: if (feature = x) and (feature= x, etc.) ⟹ 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) ⟹ da=task:check (5.0/1.0)
(right = p) and (max.pitch <= 203.87) ⟹ 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) ⟹ da=task:yna (60.0/22.0)
(reply_to = task;ynq;t_give) ⟹ da=task:yna (2.0/0.0)
(reply_to = task;ynq;t_grab) ⟹ da=task:yna (2.0/0.0)
(reply_to = task;ynq;t_release) ⟹ 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) ⟹ 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) ⟹ da=task:inf.agree (8.0/1.0)
(yeah = p) and (energy >= 56.862651) and (mean.pitch <= 144.971) ⟹ da=task:inf.agree (9.0/2.0)
(dss_reply <= -0.359985) and (sure = p) and (max.pitch <= 187.065) ⟹ da=task:inf.agree (8.0/0.0)
(yeah = p) and (U3 = turn:t_keep;time:stal) ⟹ 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) ⟹ da=task:inf.warn (7.0/3.0)
(because = p) ⟹ da=task:inf.just (33.0/7.0)
(cause = p) ⟹ 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) ⟹ 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) ⟹ da=task:suggest (38.0/11.0)
(duration >= 2.12) and (reply_to = _) and (might = p) ⟹ 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) ⟹ da=au_f:au_f_p_ex (168.0/24.0)
(dss_reply <= -0.039917) and (duration <= 1.08) and (okay = p) ⟹ da=au_f:au_f_p_ex (84.0/8.0)
(dss_reply <= -0.039978) and (break <= 1) and (mmhmm = p) ⟹ da=au_f:au_f_p_ex (34.0/1.0)
(dss_reply <= -0.039978) and (break <= 3) and (voclaugh = p) ⟹ da=au_f:au_f_p_ex (25.0/2.0)
(okay = p) and (energy <= 56.617891) and (duration >= 1.16) ⟹ 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) ⟹ da=au_f:au_f_p_ex;turn:t_give (83.0/11.0)
(within = turn:t_keep;time:stal) and (energy <= 50.235299) ⟹ 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*)

Dialogue 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: Dialogue Structuring; GP:INFORM; FT: *ds:inf*)
(1036:44-1037:68) B: And that's the end of the meeting (dimension: Dialogue 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*)