Incremental dialogue act understanding

Volha Petukhova
Tilburg Center for Creative Computing
Tilburg University, The Netherlands,
v.petukhova@uvt.nl

Harry Bunt
Tilburg Center for Creative Computing
Tilburg University, The Netherlands,
harry.bunt@uvt.nl

Abstract

This paper presents a machine learning-based approach to the incremental understanding of dialogue utterances, with a focus on the recognition of their communicative functions. A token-based approach combining the use of local classifiers, which exploit local utterance features, and global classifiers which use the outputs of local classifiers applied to previous and subsequent tokens, is shown to result in excellent dialogue act recognition scores for unsegmented spoken dialogue. This can be seen as a significant step forward towards the development of fully incremental, on-line methods for computing the meaning of utterances in spoken dialogue.

1 Introduction

When reading a sentence in a text, a human language understander obviously does not wait trying to understand what he is reading until he has come to the end of the sentence. Similarly for participants in a spoken conversation. There is overwhelming psycholinguistic evidence that human understanders construct syntactic, semantic, and pragmatic hypotheses on the fly, while receiving the written or spoken input. Dialogue phenomena such as backchannelling (providing feedback while someone else is speaking), the completion of a partner utterance, and requests for clarification that overlap the utterance of the main speaker, illustrate this. Evidence from the analysis of nonverbal behaviour in multimodal dialogue lends further support to the claim that human understanding works incrementally, as input is being received. Dialogue participants start to perform certain body movements and facial expressions that are perceived and interpreted by others as dialogue acts (such as head nods, smiles, frowns) while another participant is speaking, see e.g. Petukhova and Bunt (2009). As another kind of evidence, eye-tracking experiments by Tanenhaus et al. (1995), Sedivy et al. (1999) and Sedivy (2003) showed that definite descriptions are resolved incrementally when the referent is visually accessible.

Traditional models of language understanding for dialogue systems, by contrast, are pipelined, modular, and operate on complete utterances. Typically, such a system has an automatic speech recognition module, a language understanding module responsible for syntactic and semantic analysis, an interpretation manager, a dialogue manager, a natural language generation module, and a module for speech synthesis. The output of each module is the input for another. The language understanding module typically performs the following tasks: (1) *segmentation*: identification of relevant segments in the input, such as sentences;(2) *lexical analysis*: lexical lookup, possibly supported by morphological processing, and by additional resources such as WordNet, VerbNet, or lexical ontologies; (3) *parsing*: construction of syntactic interpretations; (4) *semantic analysis*: computation of propositional, referential, or action-related content; and (5) *pragmatic analysis*: determination of speaker intentions.

Of these tasks, lexical analysis, being concerned with local information at word level, can be done for each word as soon as it has been recognized, and is naturally performed as an incremental part of utterance processing, but syntactic, semantic and pragmatic analysis are traditionally performed on complete utterances. Tomita's pioneering work in left-to-right syntactic parsing has shown that incremental parsing can be much more efficient and of equal quality as the parsing of complete utterances (Tomita (1986)). Computational approaches to incremental semantic and pragmatic interpretation have

been less successful (see e.g. Haddock (1989); Milward and Cooper (2009)), but work in computational semantics on the design of underspecified representation formalisms has shown that such formalisms, developed originally for the underspecified representation of quantifier scopes, can also be applied in situations where incomplete input information is available (see e.g. Bos (2002); Bunt (2007), Hobbs (1985), Pinkal (1999)) and as such hold a promise for incremental semantic interpretation.

Pragmatic interpretation, in particular the recognition of a speaker's intentions in incoming dialogue utterances, is another major aspect of language understanding for dialogue systems. Computational modelling of dialogue behaviour in terms of dialogue acts aims to capture speaker intentions in the communicative functions of dialogue acts, and offers an effective integration with semantic content analysis through the information state update approach (Poesio and Traum (1998)). In this approach, a dialogue act is viewed as having as its main components a communicative function and a semantic content, where the semantic content is the referential, propositional, or action-related information that the dialogue act addresses, and the communicative function defines how an understander's information state is to be updated with that information.

Evaluation of a non-incremental dialogue system and its incremental counterpart reported in Aist et al. (2007) showed that the latter is faster overall than the former due to the incorporation of pragmatic information in early stages of the understanding process. Since users formulate utterances incrementally, partial utterances may be available for a substantial amount of time and may be interpreted by the system. An incremental interpretation strategy may allow the system to respond more quickly, by minimizing the delay between the time the user finishes and the time the utterance is interpreted DeVault and Stone (2003).

This suggests that a dialogue system performance may benefit from reliable partial processing of input. This paper is concerned with the automatic recognition of dialogue acts based on partially available input and shows that in order to arrive at the best output prediction two different classification strategies are needed: (1) local classification that is based on features observed in dialogue behaviour and that can be extracted from the annotated data; and (2) global classification that takes the locally predicted context into account.

This paper is structured as follows. In Section 2 we will outline performed experiments describing the data, tagset, features, algorithms and evaluation metrics that have been used. Section 3 reports on the experimental results, applying a variety of machine learning techniques and feature selection algorithms, to assess the automatic recognition and classification of dialogue acts using simultaneous incremental segmentation and dialogue act classification. In Section 4 we discuss strategies in management and correction of the output of local classifies. Section 5 concludes.

2 Incremental understanding experiments

2.1 Related work

Nakano et al. (Nakano et al. (1999)) proposed a method for the incremental understanding of utterances whose boundaries are not known. The *Incremental Sentence Sequence Search* (ISSS) algorithm finds plausible boundaries of utterances, called significant utterances (SUs), which can be a full sentence or a subsentential phrase, such as a noun phrase or a verb phrase. Any phrase that can change the belief state is defined as a SU. In this sense an SU corresponds more or less with what we call a 'functional segment', which is defined as a minimal stretch of behaviour that has a communicative function (see Bunt et al. (2010)). ISSS maintains multiple possible belief states, and updates these each time a word hypothesis is input. The ISSS approach does not deal with the multifunctionality of segments, however, and does not allow segments to overlap.

Lendvai and Geertzen (Lendvai and Geertzen (2007)) proposed *token-based* dialogue act segmentation and classification, which was worked out in more detail in Geertzen (2009). This approach takes dialogue data that is not segmented into syntactic or semantic units, but operates on the transcribed speech as a stream of words and other vocal signs (e.g. laughs), including disfluent elements (e.g. abandoned

Dimension	Frequency	General-purpose function	Frequency
Task	31.8	PropositionalQuestion	5.8
Auto-Feedback	20.5	Set Question	2.3
Allo-Feedback	0.7	Check Question	3.3
Turn Management	50.2	Propositional Answer	9.8
Social Obligation Management	0.5	Set Answer	3.9
Discourse Structuring	2.8	Inform	11.7
Own Communication Management	10.3	InformRhetorical	21.9
Time Management	26.7	Instruct	0.3
Partner Communication Management	0.3	Suggest	10.1
Contact Management	0.1	Request	5.6

Table 1: *Distribution of functional tags across dimensions and general-purpose functions for the AMI corpus (in %).*

or interrupted words). Segmentation and classification of dialogue acts are performed simultaneously in one step. Geertzen (2009) reports on classifier performance on this task for the DIAMOND data[1] using DIT^{++} labels. The success scores in terms of F-scores range from 47.7 to 81.7. It was shown that performing segmentation and classification together results in better segmentation, but affects the dialogue act classification negatively.

The incremental dialogue act recognition system proposed here takes the token-based approach for building classifiers for the recognition (segmentation and classification) of multiple dialogue acts for each input token, and adopts the ISSS idea for information-state updates based on partial input interpretation.

2.2 Tagset

The data selected for the experiments was annotated with the DIT^{++} tagset Release 4[2]. The DIT taxonomy distinguishes 10 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, introducing, changing or closing a topic (*Dialogue Structuring*), and conditions that trigger dialogue acts by social convention (*Social Obligations Management*), see Table 1.

For each dimension, at most one communicative function can be assigned, which is either a function that can occur in this dimension alone (a *dimension-specific* (DS) function) or a function that can occur in any dimension (a *general-purpose* (GP) function). Dialogue acts with a DS communicative function are always concerned with a particular type of information, such as a Turn Grabbing act, which is concerned with the allocation of the speaker role, or a Stalling act, which is concerned with the timing of utterance production. GP functions, by contrast, are not specifically related to any dimension in particular, e.g. one can ask a question about any type of semantic content, provide an answer about any type of content, or request the performance of any type of action (such as *Could you please close the door* or *Could you please repeat that*). These communicative functions include Question, Answer, Request, Offer, Inform, and many other familiar core speech acts.

The tagset used in these studies contains 38 dimension-specific functions and 44 general-purpose functions. A tag consists either of a pair consisting of a communicative function (CF) and the addressed dimension (D).

[1]For more information see Geertzen,J., Girard,Y., and Morante,R. 2004. The DIAMOND project. Poster at the 8th Workshop on the Semantics and Pragmatics of Dialogue (CATALOG 2004).

[2]For more information about the tagset and the dimensions that are identified, please visit:`http://dit.uvt.nl/` or see Bunt (2009).

Speaker	Token	Task	Auto-F.	Allo-F.	TurnM.	TimeM.	ContactM.	DS	OCM	PCM	SOM
B	it	B;inf	O	O	O	O	O	O	O	O	O
B	has	I:inf	O	O	O	O	O	O	O	O	O
B	to	I:inf	O	O	O	O	O	O	O	O	O
B	look	I:inf	O	O	O	O	O	O	O	O	O
B	you	O	O	B:check	O	O	O	O	O	O	O
B	know	O	O	E:check	O	O	O	O	O	O	O
B	cool	I:inf	O	O	O	O	O	O	O	O	O
D	mmhmm	O	BE:positive	O	O	O	O	O	O	O	O
B	and	I:inf	O	O	BE:t_keep	O	O	O	O	O	O
B	gimmicky	E:inf	O	O	O	O	O	O	O	O	O

Figure 1: *Segment boundaries and dialogue act label encoding in different dimensions.*

2.3 Features and data encoding

In the recognition experiments we used data from the AMI meeting corpus[3]. For training we used three annotated AMI meetings that contain 17,335 tokens forming 3,897 functional segments. The distribution of functional tags across dimensions is given in Table 1.

Features extracted from the data considered here relate to *dialogue history*: functional tags of the 10 previous turns; *timing*: token *duration* and *floor-transfer offset*[4] computed in milliseconds; *prosody*: minimum, maximum, mean, and standard deviation for pitch (F0 in Hz), energy (RMS), voicing (fraction of locally unvoiced frames and number of voice breaks) and speaking rate (number of syllables per second)[5]; and *lexical information*: token occurrence, bi- and trigram of those tokens. In total, 1,668 features are used for the AMI data.

To be able to identify segment boundaries, we assign to each token its communicative function label and indicate whether a token starts a segment (B), is inside a segment (I), ends a segment (E), is outside a segment (O), or forms a functional segment on its own (BE). Thus, the class labels consist of a segmentation prefix (IBOE) and a communicative function label, see example in Figure 1.

2.4 Classifiers and evaluation metrics

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

As a probabilistic classifier we used *Bayes Nets*. This classifier estimates probabilities rather than produce predictions, which is often more useful because this allows us to rank predictions. Bayes Nets estimate the conditional probability distribution on the values of the class attributes given the values of the other attributes.

As a rule induction algorithm we chose *Ripper* (Cohen (1995)). The advantage of a rule inducer is that the regularities discovered in the data are represented as human-readable rules.

The results of all experiments were obtained using 10-fold cross-validation.[7] As a baseline it is common practice to use the majority class tag, but for our data sets such a baseline is not very useful because of the relatively low frequencies of the tags in some dimensions. Instead, we use a baseline

[3]The Augmented Multi-party Interaction meeting corpus consists of multimodal task-oriented human-human multi-party dialogues in English, for more information visit (`http://www.amiproject.org/`

[4]Difference between the time that a turn starts and the moment the previous turn ends.

[5]These features were computed using the PRAAT tool[6]. We examined both raw and normalized versions of these features. Speaker-normalized features were obtained by computing z-scores (z = (X-mean)/standard deviation) for the feature, where mean and standard deviation were calculated from all functional segments produced by the same speaker in the dialogues. We also used normalizations by first speaker turn and by previous speaker turn.

[7]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 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.

that is based on a single feature, namely, the tag of the previous dialogue utterance (see Lendvai et al. (2003))).

Several metrics have been proposed for the evaluation of a classifier's performance: error metrics and performance metrics. The word-based error rate metric, introduced in Ang et al. (2005), measures the percentage of words that were placed in a segment perfectly identical to that in the reference. The dialogue act based metric (DER) was proposed in Zimmermann et al. (2005). In this metric a word is considered to be correctly classified if and only if it has been assigned the correct dialogue act type and it lies in exactly the same segment as the corresponding word of the reference. We will use the combined DER_{sc} error metric to evaluate joint segmentation (s) and classification (c):

$$DER_{sc} = \frac{Tokens\ with\ wrong\ boundaries\ and/or\ function\ class}{total\ number\ of\ tokens} \times 100$$

To assess the quality of classification results, the standard F-score metric is used, which represents the balance between precision and recall.

3 Classification results

Dialogue utterances are often multifunctional, having a function in more than one dimension (see e.g. Bunt (2010)). This makes dialogue act recognition a complex task. Splitting up the output structure may make the task more manageable; for instance, a popular strategy is to split a multi-class learning task into several binary learning tasks. Sometimes, however, learning of multiple classes allows a learning algorithm to exploit the interactions among classes. We will combine these two strategies. We have built in total 64 classifiers for dialogue act recognition for the AMI data. Some of the tasks were defined as binary ones, e.g. the dimension recognition task, others are multi-class learning tasks.

We first trained classifiers to recognize the boundaries of a segment and its communicative functions (joint multi-class learning task) per dimension, see Table 2.

	BL		BayesNet		Ripper	
Dimensions	F_1	DER_{sc}	F_1	DER_{sc}	F_1	DER_{sc}
Task	32.7	51.2	52.1	48.7	**66.7**	42.6
Auto-Feedback	43.2	84.4	**62.7**	33.9	60.1	45.6
Allo-Feedback	70.2	59.5	**73.7**	35.1	71.3	49.1
Turn Management:initial	34.2	95.2	**57.0**	58.4	54.3	81.3
Turn Management:close	33.3	92.7	**54.2**	46.9	49.3	87.3
Time Management	43.7	96.5	**64.5**	46.1	61.4	53.1
Discourse Structuring	41.2	35.1	**72.7**	19.9	50.2	30.9
Contact Management	59.9	53.2	71.4	49.9	**83.3**	37.2
Own Communication Management	36.5	87.9	**68.3**	51.3	58.3	76.8
Partner Communication Management	49.5	59.0	**58.5**	45.5	51.4	58.7
Social Obligation Management	34.5	47.5	**86.5**	35.9	83.3	44.3

Table 2: Overview of F-scores and DER_{sc} for the baseline (BL) and the classifiers for joint segmentation and classification for each DIT^{++} dimension, for the data of the AMI corpus.

The results show that both classifiers outperform the baseline by a broad margin. The Bayes Nets classifier marginally outperforms the Ripper rule inducer, but shows no significant differences in overall performance. Though the results obtained are quite encouraging, the performance on the joint segmentation and classification task does not outperforms the two-step segmentation and classification task reported in Geertzen et al. (2007). There is a drop in F-scores compared to the results reported by Geertzen et al. (2007), which is explained by the fact that recall was quite low. This means that the classifiers missed a lot of relevant cases. Looking more closely at the predictions made by the classifiers, we noticed that beginnings and endings of many segments were not found. For example, the beginnings of Set Questions are identified with perfect precision (100%), but about 60% of the segment beginnings were not found. The reason that the classifiers still show a reasonable performance is that most tokens occur

inside segments and are better classified, e.g. the inside-tokens of Set Questions are classified with high precision (83%) and reasonably high recall scores (76%). Still, this is rather worrying, since the correct identification of, in particular, the start of a relevant segment is crucial for future decisions. These observations led us to the conclusion that the search space and the number of initially generated hypotheses for classifiers should be reduced, and we split the classification task in such a way that a classifier needs to learn one particular type of communicative function.

We trained a classifier for each general-purpose and dimension-specific function defined in the DIT^{++} taxonomy, and observed that this has the effect that the various classifiers perform significantly better. These functions were learned (1) in isolation; (2) as semantically related functions together, e.g. all information-seeking functions (all types of questions) or all information-providing functions (all answers and all informs). Both the recognition of communicative functions and that of segment boundaries improves significantly. Table 3 gives an overview of the overall performance (best obtained scores) of the trained classifiers after splitting the learning task.

	BL		BayesNet		Ripper	
Classification task	F_1	DER_{sc}	F_1	DER_{sc}	F_1	DER_{sc}
General-purpose functions						
Propositional Questions	47.0	39.1	**94.9**	3.9	75.8	23.5
Check Questions	43.8	56.4	**68.5**	19.6	61.3	33.1
Set Questions	44.8	52.1	74.1	18.6	**76.3**	17.7
Inform	45.8	39.9	**79.8**	18.7	66.5	30.5
Inform Rhetorical	37.2	38.9	**69.1**	13.4	68.7	23.9
Agreement	41.3	79.1	**72.1**	12.6	71.6	60.2
Propositional Answer	32.0	77.8	**66.8**	26.1	52.2	53.8
Set Answer	44.3	54.2	**77.5**	13.2	57.3	44.1
Suggest	45.8	38.4	**65.6**	17.3	48.8	35.6
Request	45.8	49.3	**75.8**	14.5	50.3	36.9
Instruct	46.3	49.3	**60.5**	14.5	46.3	36.9
Dimension-specific functions						
Auto-Feedback	57.1	23.5	**78.8**	13.2	66.7	15.5
Allo-Feedback	89.3	4.4	**95.1**	2.9	94.3	3.9
Turn Management:initial	24.8	21.9	**72.8**	7.4	46.3	10.7
Turn Management:close	30.7	64.9	**62.0**	22.5	54.7	39.6
Time management	68.3	32.3	82.4	13.7	**92.8**	11.4
Discourse Structuring	40.7	13.6	72.6	2.5	**74.5**	1.7
Contact Management	21.4	48.6	89.2	5.7	**92.3**	3.6
Own Communication Management	26.7	48.6	**78.0**	11.6	68.1	20.0
Partner Communication Management	33.4	18.2	77.8	8.5	**88.9**	6.5
Social Obligation Management	60.0	18.7	88.9	8.3	**90.1**	5.5

Table 3: Overview of F-scores and DER_{sc} for the baseline (BL) and the classifiers upon joint segmentation and classification task for each DIT^{++} communicative function or cluster of functions. (Best scores indicated by numbers in bold face.)

Segments having a general-purpose functions may address any of the ten DIT dimensions. The task of dimension recognition can be approached in two ways. One approach is to learn segment boundaries, communicative function label and dimension in one step (e.g. the class label *B:task;inform*). This task is very complicated, however. First, it leads to data which are high dimensional and sparse, which will have a negative influence on the performance of the trained classifiers. Second, in many cases the dimension can be recognized reliably only with some delay; for the first few segment tokens it is often impossible to say what the segment is about. For example:

(1) 1. What do you think who we're aiming this at?
2. What do you think we are doing next?
3. What do you think Craig?

The three Set Questions in (1) start with exactly the same words, but they address different dimensions: Question 1 is about the Task (in AMI - the design the television remote control); Question 2 serves the

purpose of Discourse Structuring; and Question 3 elicits feedback.

Another approach is to first recognize segment boundaries and communicative function, and define dimension recognition as a separate classification task.

Tokens	SetQuestion		Task		Auto-F.		TurnM.		Complex label (BIOE:D;CF)	
	label	p	label	p	label	p	label	p	label	p
what	B:setQ	0.85	O	0.71	O	1	O	0.68	O	0.933
you	I:setQ	1	task	0.985	O	1	B:give	0.64	O	0.869
guys	I:setQ	1	task	0.998	O	1	E:give	0.66	O	0.937
have	I:setQ	1	task	0.997	O	1	O	1	I:task;setQ	0.989
already	I:setQ	1	task	0.996	O	1	O	0.99	I:task;setQ	0.903
received	I:setQ	1	task	0.987	O	1	O	1	I:task;setQ	0.813
um	O	0.93	O	0.89	O	1	BE:keep	0.99	O	0.982
in	I:setQ	1	task	0.826	O	1	O	0.89	I:task;setQ	0.875
your	I:setQ	1	task	0.996	O	1	O	0.99	I:task;setQ	0.948
mails	E:setQ	0.99	task	0.987	O	1	O	1	E:task;setQ	0.948

Figure 2: *Predictions with indication of confidence scores (highest p class probability selected) for each token assigned by five trained classifiers simultaneously.*

We tested both strategies. The F-scores for the joint learning of complex class labels range from 23.0 (DER_{sc} = 68.3) to 45.3 (DER_{sc} = 63.8). For dimension recognition as a separate learning task the F-scores are significantly higher, ranging from 70.6 to 97.7. The scores for joint segmentation and function recognition in the latter case are those listed in Table 3. Figure 2 gives an example of predictions made by five classifiers for the input *what you guys have already received um in your mails*.

4 Managing local classifiers

4.1 Global classification and global search

As shown in the previous section, given a certain input we obtain all possible output predictions (hypotheses) from local classifiers. Some predictions are false, but once a local classifier has made a decision it is never revisited. It is therefore important to base the decision on dialogue act labels not only on local features of the input, but to take other parts of the output into account as well. For example, the partial output predicted so far, i.e. the history of previous predictions, may be taken as features for the next classification step, and helps to discover and correct errors. This is known as 'recurrent sliding window strategy' (see Dietterich (2002)) when the true values of previous predictions are used as features. This approach suffers from the label bias problem, however, when a classifier overestimates the importance of certain features, and moreover does not apply in a realistic situation, since the true values of previous predictions are not available to a classifier in real time. A solution proposed by Van den Bosch (1997) is to apply adaptive training using the *predicted* output of previous steps as features.

We trained higher-level classifiers (often referred to as 'global') that have, along with features extracted locally from the input data as described above, the partial output predicted so far from all local classifiers. We used five previously predicted class labels, assuming that long distance dependencies may be important, and taking into account that the average length of a functional segment in our data is 4.4 tokens. Table 4 gives an overview of the results of applying these global classifiers. We see that the global classifiers make more accurate predictions than the local classifiers, showing an improvement of about 10% on average. The classifiers still make some incorrect predictions, because the decision is sometimes based on incorrect previous predictions. An optimized global search strategy may lead to further improvements of these results.

A strategy to optimize the use of output hypotheses, is to perform a global search in the output space looking for best predictions. Our classifiers do not just predict the most likely class for an instance, but also generate a distribution of output classes. Class distributions can be seen as confidence scores of all predictions that led to a certain state. Our confidence models are constructed based on token level information given the dialogue left-context (i.e. dialogue history, wording of the previous and

Classification task	BayesNet		Ripper	
	F_1	DER_{sc}	F_1	DER_{sc}
Task	65.3	14.9	**79.1**	21.8
Auto-Feedback	72.9	8.1	**77.8**	7.2
Allo-Feedback	67.7	10.9	**74.2**	9.5
Turn Management:initial	**72.2**	11.5	69.5	11.4
Turn Management:close	82.7	5.0	**83.0**	4.9
Time Management	70.0	3.0	**73.5**	2.1
Discourse Structuring	**72.3**	4.9	63.7	3.6
Contact Management	79.1	4.5	**84.3**	4.6
Own Communication Management	66.0	2.4	**68.3**	2.3
Partner Communication Management	**63.2**	7.8	59.5	11.4
Social Obligation Management	**88.4**	0.9	81.6	1.7

Table 4: Overview of F-scores and DER_{sc} of the global classifiers for the AMI data based on added previous predictions of local classifiers.

currently produced functional segment). This is particular useful for dialogue act recognition because the recognition of intentions should be based on the system's understanding of discourse and not just on the interpretation of an isolated utterance. Searching the (partial) output space for the best predictions is not always the best strategy, however, since the highest-ranking predictions are not always correct in a given context. A possible solution to this is to postpone the prediction until some (or all) future predictions have been made for the rest of the segment. For training, the classifier then uses not only previous predictions as additional features, but also some or all future predictions of local classifiers (till the end of the current segment or to the beginning of the next segment, depending on what is recognized). This forces the classifier to not immediately select the highest-ranking predictions, but to also consider lower-ranking predictions that could be better in the context of the rest of the sequence.

Classification task	BayesNet		Ripper	
	F_1	DER_{sc}	F_1	DER_{sc}
Task	82.6	9.5	**86.1**	8.3
Auto-Feedback	81.9	1.9	**95.1**	0.6
Allo-Feedback	**96.3**	0.6	95.7	0.5
Turn Management:initial	**85.7**	1.5	81.5	1.6
Turn Management:close	90.9	3.8	**91.2**	3.6
Time management	90.4	2.4	**93.4**	1.7
Discourse Structuring	**82.1**	1.7	78.3	1.8
Contact Management	87.9	1.2	**94.3**	0.6
Own Communication Management	78.4	2.2	**81.6**	2.0
Partner Communication Management	**71.8**	2.4	70.0	4.6
Social Obligation Management	98.6	0.4	98.6	0.5

Table 5: Overview of F-scores and DER_{sc} of global classifiers for the AMI data per DIT^{++} dimension.

The results show the importance of optimal global classification for finding the best output prediction.

We performed similar experiments on the English MapTask data[8] and obtained comparable results, where F-scores on the global classification task range from 66.7 for Partner Communication Management and Discourse Structuring to 79.7 for Task and 91.2 for Allo-Feedback. For the MapTask corpus the performance of human annotators on segmentation and classification has been assessed; standard kappa scores reported in Bunt et al. (2007) range between 0.92 and 1.00, indicating near perfect agreement between two expert annotators[9].

[8]For more information about the MapTask corpus see `http://www.hcrc.ed.ac.uk/maptask/`

[9]Note, however, that a slightly simplified version of the DIT^{++} tagset has been used here, called the LIRICS tagset, in which the five DIT levels of processing in the Auto- and Allo-Feedback dimensions were collapsed into one.

5 Conclusions and future research

The incremental construction of input interpretation hypotheses is useful in a language understanding system, since it has the effect that the understanding of a relevant input segment is already nearly ready when the last token of the segment is received; when a dialogue act is viewed semantically as a recipe for updating an information state, this means that the specification of the update operation is almost ready at that moment, thus allowing an instantaneous response from the system. It may even happen that the confidence score of a partially processed input segment is that high, that the system may decide to go forward and update its information state without waiting until the end of the segment, and prepare or produce a response based on that update. Of course, full incremental understanding of dialogue utterances includes not only the recognition of communicative functions, but also that of semantic content. However, many dialogue acts have no or only marginal semantic content, such as turn-taking acts, backchannels (*m-hm*) and other feedback acts (*okay*), time management acts (*Just a moment*), apologies and thankings and other social obligation management acts, and in general dialogue acts with a dimension-specific function; for these acts the proposed strategy can work well without semantic content analysis, and will increase the system's interactivity significantly. Moreover, given that the average length of a functional segment in our data is no more than 4.4 tokens, the semantic content of such a segment tends not to be very complex, and its construction therefore does not seem to require very sophisticated computational semantic methods, applied either in an incremental fashion (see e.g. Aist et al. (2007) and DeVault and Stone (2003)) or to a complete segment.

Interactivity is however not the sole motivation for incremental interpretation. The integration of pragmatic information obtained from the dialogue act recognition module, as proposed here, at early processing stage can be beneficially used by the incremental semantic parser (but also syntactic parser module). For instance, information about the communicative function of the incoming segment at early processing stage can defuse a number of ambiguous interpretations, e.g. used for the resolution of many anaphoric expressions. A challenge for future work is to integrate the incremental recognition of communicative functions with incremental syntactic and semantic parsing, and to exploit the interaction of syntactic, semantic and pragmatic hypotheses in order to understand incoming dialogue segments incrementally in an optimally efficient manner.

Acknowledgments

This research was conducted within the project 'Multidimensional Dialogue Modelling', sponsored by the Netherlands Organisation for Scientific Research (NWO), under grant reference 017.003.090. We are also very thankful to anonymous reviewers for their valuable comments.

References

Aist, G., J. Allen, E. Campana, C. Gomez Gallo, S. Stoness, M. Swift, and M. K. Tanenhaus (2007). Incremental understanding in human-computer dialogue and experimental evidence for advantages over nonincremental methods. In *Proceedings of the 11th Workshop on the Semantics and Pragmatics of Dialogue*, Trento, Italy, pp. 149–154.

Ang, J., Y. Liu, and E. Shriberg (2005). Automatic dialog act segmentation and classification in multiparty meetings. In *Proceedings of the ICASSP*, Volume vol. 1, Philadelphia, USA, pp. 10611064.

Bos, J. (2002). *Underspecification and resolution in discourse semantics. PhD Thesis*. Saarbrücken: Saarland University.

Bunt, H. (2007). Semantic underspecification: which techniques for what purpose? In *Computing Meaning, Vol. 3*, pp. 55–85. Dordrecht: Springer.

Bunt, H. (2009). The DIT++ taxonomy for functional dialogue markup. In *Proceedings of the AAMAS 2009 Workshop 'Towards a Standard Markup Language for Embodied Dialogue Acts' (EDAML 2009)*, Budapest.

Bunt, H. (2010). Multifunctionality in dialogue and its interpretation. *Computer, Speech and Language, Special issue on dialogue modeling*.

Bunt, H., J. Alexandersson, J. Carletta, J.-W. Choe, A. Fang, K. Hasida, K. Lee, V. Petukhova, A. Popescu-Belis, L. Romary, C. Soria, and D. Traum (2010). *Language resource management – Semantic annotation framework – Part 2: Dialogue acts. ISO DIS 24617-2*. Geneva: ISO Central Secretariat.

Bunt, H., V. Petukhova, and A. Schiffrin (2007). Lirics deliverable d4.4. multilingual test suites for semantically annotated data. Available at `http://lirics.loria.fr`.

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

DeVault, D. and M. Stone (2003). Domain inference in incremental interpretation. In *Proceedings of the Workshop on Inference in Computational Semantics*, INRIA Lorraine, Nancy, France.

Dietterich, T. (2002). Machine learning for sequential data: a review. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, pp. 15–30.

Geertzen, J. (2009). *Dialogue act recognition and prediction: exploration in computational dialogue modelling*. The Netherlands: Tilburg University.

Geertzen, J., V. Petukhova, and H. Bunt (2007, September). A multidimensional approach to utterance segmentation and dialogue act classification. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, Antwerp, Belgium, pp. 140–149. Association for Computational Linguistics.

Haddock, N. (1989). Computational models of incremental semantic interpretation. *Language and Cognitive Processes Vol. 14 (3)*, SI337–SI380.

Hobbs, J. (1985). Ontological promiscuity. In *Proceedings 23rd Annual Meeting of the ACL*, Chicago, pp. 61–69.

Lendvai, P., v. d. A. Bosch, and E. 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*, Budapest.

Lendvai, P. and J. Geertzen (2007). Token-based chunking of turn-internal dialogue act sequences. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, Antwerp, Belgium, pp. 174–181.

Milward, D. and R. Cooper (2009). Incremental interpretation: applications, theory, and relationship to dynamic semantics. In *Proceedings COLING 2009, Kyoto, Japan*, pp. 748–754.

Nakano, M., N. Miyazaki, J. Hirasawa, K. Dohsaka, and T. Kawabata (1999). Understanding unsegmented user utterances in real-time spoken dialogue systems. In *Proceedings of the 37th Annual Conference of the Association of Computational Linguistics, ACL*, pp. 200–207.

Petukhova, V. and H. Bunt (2009). Who's next? speaker-selection mechanisms in multiparty dialogue. In *Proceedings of the Workshop on the Semantics and Pragmatics of Dialogue*, Stockholm,, pp. 19–26.

Pinkal, M. (1999). On semantic underspecification. In *Computing Meaning, Vol. 1*, pp. 33–56. Dordrecht: Kluwer.

Poesio, M. and D. Traum (1998). Towards an Axiomatization of Dialogue Acts. In *Proceedings of the Twente Workshop on the Formal Semantics and Pragmatics of Dialogue*, Twente, pp. 309–347.

Sedivy, J. (2003). Pragmatic versus form-based accounts of referential contrast: Evidence for effects of informativity expectations. *Journal of Psycolinguistic Research 32(1)*, 3–23.

Sedivy, J., M. Tanenhaus, C. Chambers, and G. Carlson (1999). Achieving incremental semantic interpretation through contextual representation. *Cognition 71*, 109–147.

Tanenhaus, M., M. Spivey-Knowlton, K. Eberhard, and J. Sedivy (1995). Intergration of visual and linguistic information in spoken language comprehension. *Science 268*, 1632–1634.

Tomita, M. (1986). *Efficient parsing for natural language*. Dordrecht: Kluwer.

Van den Bosch, A. (1997). *Learning to pronounce written words: A study in inductive language learning. PhD thesis*. The Netherlands: Maastricht University.

Zimmermann, M., Y. Lui, E. Shriberg, and A. Stolcke (2005). Toward joint segmentation and classification of dialog acts in multiparty meetings. In *Proceedings of the Multimodal Interaction and Related Machine Learning Algorithms Workshop (MLMI05)*, pp. 187–193. Springer.