Self-Organizing Dialogue Management

Kristiina Jokinen, Topi Hurtig, Kevin Hynnä, Kari Kanto, Mauri Kaipainen, and Antti Kerminen

Media Lab
University of Art and Design Helsinki
Hämeentie 135 C 00560 Helsinki FINLAND
kristiina.jokinen@uiah.fi

Abstract

In this paper, we present our approach to dialogue management in the spoken dialogue system that is being developed within the project Interact. Compared to traditional approaches, our dialogue manager will support the system's adaptivity and flexibility with the help of two design decisions: an agent-based architecture and the use of neural network models. Our experiments focus on word-based dialogue act recognition using the LVQ classification algorithm in a corpus of information-seeking dialogues, and we compare the results with a simple bag-of-words approach. We also report our studies of clustering the input data into necessary and meaningful categories using self-organizing maps.

1 Introduction

One of the main tasks of the dialogue manager component in spoken dialogue systems is to recognize communicative intentions behind the user's utterances in order to react to the natural language requests in an appropriate way (e.g. Alexandersson *et al.* 2000, Allwood *et al.* 2000). The intentions, often referred to as dialogue acts, serve, together with other contextual information such as the topic of the utterance and the underlying task, as the basis for the system action, whether it is information retrieving from a database or a question to clarify an insufficiently specified request.

Many approaches to dialogue management and dialogue act recognition are available, ranging from symbolic reasoning using various types of plans (TRIPS, Allen *et al.* 2000, VERBMOBIL, Alexandersson *et al.* 2000, TRINDI, Larsson *et al.* 2000), logical inference (ARTIMIS, Sadek *et al.* 1997), and focus (LINLIN, Jönsson 1997, Jokinen 1996), to statistics (Nagata and Morimoto 1994, VERBMOBIL, Reithinger and Maier 1995), reinforcement learning (Litman *et al.* 2000), and neural networks (DISCERN, Miikkulainen 1993, SCREEN, Wermter and Weber 1997, DIA-MOLE, Möller 1996).

However, in practice dialogue act recognition is usually based on more modest keyword spotting, and the dialogue manager uses a few symbolic update rules. The reason lies in the complex syntactic structure of utterances and the intrinsic fuzziness of the dialogue act types. Recognition accuracy is further hurt by spoken language characteristics such as errors, corrections, feedback markers and hesitations, and mere ASR errors often render lengthy spoken input as garbage.

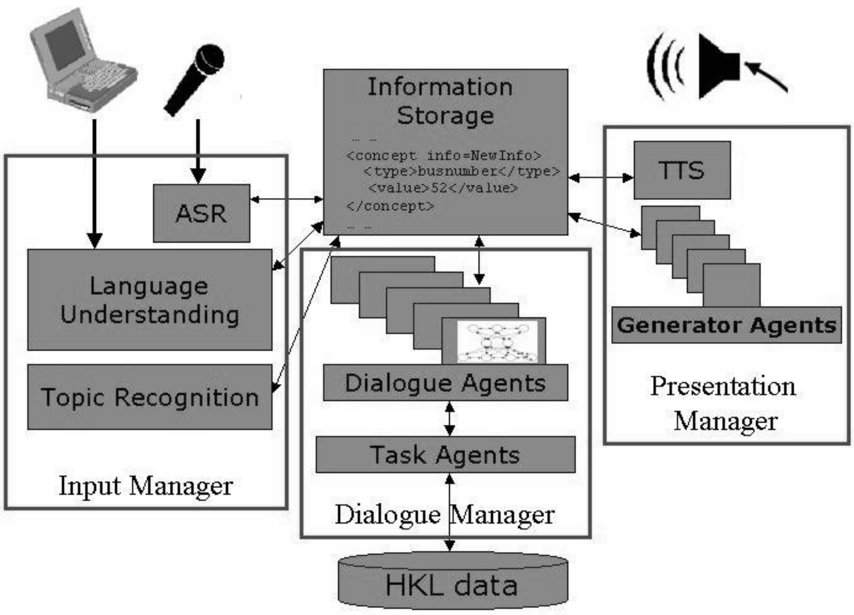
In Interact, a collaboration project between Finnish universities and IT companies (Jokinen 2001), we aim at developing a generic dialogue model that would overcome rigidity of practical dialogue systems and enable users to interact with applications in a natural way. The point of departure is to experiment with tools and methods that license the system to approach various problematic situations flexibly. For instance, in dialogue act recognition, we investigate possibilities to learn meaningful categories for the purposes of practical systems from naturally occurring dialogues. We also look for solutions to interface design where the system is equipped with adaptive strategies that allow for handling of technical restrictions and complexity of interaction. Our dialogue manager is thus built on two design decisions: agent-based modular architecture and the use of adaptive learning techniques such as neural networks, both of which support easy accommodation to various dialogue situations, as well as quick experimentation with processing models.

In this paper we concentrate on the emergent nature of dialogue organisation and report our

studies on dialogue act recognition using only elementary data description and data-driven self-organizing maps, especially the Learning Vector Quantization method, as a classification tool. In particular, we categorise utterances into dialogue acts based on the words that occur in the utterance and their morphosyntactic features.
The paper is organised as follows. We first introduce the dialogue model and the dialogue system itself. We then describe our corpus and the SOM-method, followed by the discussion of our experiments on recognizing different types of dialogue acts. Finally, we conclude by comparing the results, and provide references for future work.

2 Dialogue model

Our view of dialogue management is based on Jokinen (1996) who defines dialogues as constructive and cooperative action in context. The speakers act by exchanging new information and constructing a shared context in which the underlying goal (which can range from a general "keep the channel open" to a specific task-goal like providing information from a database) can be achieved. As long as the goal is valid, i.e. not yet achieved or abandoned, the speakers continue their interaction, and proceed by taking turns to specify, clarify and elaborate the information exchanged.
Each action by a speaker results in a new dialogue state which is currently characterized by the agent's perception of the context: the Dialogue Act (Dact) and the Topic (Top) of the last utterance, the unfulfilled task goals (TGoal), and the last speaker. Dact describes the act that the speaker performs by a particular utterance, and Top denotes the semantic content of the utterance related to the application domain itself. Together they create a useful first approximation of the utterance meaning by abstracting over possible linguistic realisations. TGoals keep track of the application related information still necessary to fulfil the underlying task (a kind of plan), and the speaker information is needed to link the state to possible speaker characteristics. Consequently, the system's internal states are reduced to a combination of these categories, all of which form an independent source of information for the system to decide on the next move.
Our dialogue system is built on the agent-based development architecture Jaspis (Turunen and Hakulinen 2000), and it is depicted in Fig. 1. On the most general level it contains managers which handle general coordination between the system components and functional modules (such as the Input Manager, the Dialogue Manager and the Presentation Manager), and within each manager there are several agents which take care of the various tasks typical for the functional domain. The modules and agents communicate via a shared knowledge base called Information Storage, where all the information about the system state is kept.



The Dialogue Manager consists of several dialogue agents dealing with dialogue situations such as recognition of dialogue acts and task goals. The agents function in parallel, but the architecture also permits us to experiment with various competing agents for the same subtask: in this case a special evaluation agent would choose the one that best fits in the particular situation. The decision of the system's next act is determined by the dialogue controller agent that collects the information produced by the other agents, and constructs an Agenda, a list of concepts that is then transformed into a natural language expression by the generation agents (Jokinen and Wilcock 2001).

The adaptivity-oriented architecture of the dialogue manager also allows us to test different types of agents that use both neural and rule-based techniques. (We can even envisage a hierarchy of neural components analogous to DISCERN (Miikkulainen 1993), although this would use a symbolic, agent-based control structure in Jaspis). The overall design of the dialogue system, together with a parser, a generator, a speech recogniser and a synthesiser is thus an example of a tightly coupled hybrid modular architecture (Wermter and Sun 2000).

3 Dialogue corpus

As a concrete application, the project is building a dialogue system that deals with public transport time-table inquiries. Our dialogue corpus consists of spoken dialogues recorded at the Helsinki regional transport (HKL) service centre, and it contains 53 dialogues between customers and the service agent. The corpus was transcribed, manually segmented and tagged with dialogue acts. There are 2241 separate utterances and 16 dialogue act types. Table 1 gives statistics and examples of the Dacts used.

The most frequent dialogue act is *statement* which differs from the similar type *answer* in that the speaker states facts or comments rather than gives an answer to the previous question. Two different feedback types are also distinguished. *Acknowledgement* is the second most frequent act and represents feedback given by the speakers that they have understood and accepted what the partner said. It usually includes turntaking, i.e. the speaker continues with a statement or a question of her own. The act *call_to_continue* is similar but it refers to back-channelling whereby the speaker gives simultaneous acknowledgment of her understanding and encourages the partner to continue without actually taking the turn.

Dialogue Act	**freq**	**%**	**Example**
statement	527	23.5	Eiköhän se löydy sitten. *I'm sure I'll find it*
acknowledgement	389	17.4	Joo *ok, right*
question	237	10.6	Ja kauanko sinne on ajoaika? *And how long does it take to get there?*
answer	213	9.5	Se tulee noin 15 minuuttia tohon Oulunkylään. *It'll take about 15 minutes to Oulunkylä.*
confirmation	162	7.2	Suunnilleen joo. *Approximately, yes.*
opening	158	7.0	Mä oon X X hei. *Hello, my name is X X.*
check	123	5.5	Eli kuudelt lähtee ensimmäiset. *So the first ones depart at 6 o'clock*
thanking	112	5.0	Kiitoksia paljon. *Thanks a lot.*
repetition	107	4.8	Kaheksan kolkyt kolme. *At 8.33 a.m.*
ending	100	4.5	Hei. *Bye.*
call_to_continue	45	2.0	Joo-o. *Uh-huh.*
wait	23	1.0	Katsotaan, hetkinen vaan. *Let's see, just a minute.*
correction	19	0.8	Ei vaan se on edellinen se Uintikeskuksen pysäkki. *No, the Uintikeskus stop is the previous one.*
completion	10	0.4	...kymmentä joo. *...ten, right.*
request_to_repeat	10	0.4	Anteeks mitä? *Sorry?*
sigh	6	0.2	Voi kauhee. *Oh dear.*

Table 1. Distribution of dialogue acts.

We characterise dialogue acts by words and by special morphosyntactic features extracted from the word forms after they have been analysed into their base forms[1]. Besides the part-of-speech category, we use five features that seem relevant for the classification: the presence of question words, the question morpheme *-kO*, and the conditional, negation, and finite verb forms. The features are listed in Table 2.

Part of speech
Wh-word
-kO
Conditional verb form
Negation verb
Finite verb

Table 2. Morphosyntactic features.

[1] Finnish is a highly inflectional language and word morphology encodes important syntactic-semantic information.

These features are treated as properties of the utterance except, of course, the word's part-of-speech which is used to select content words for further processing (see below).

4 Self-organizing maps

We use LVQ-classification and self-organizing maps (SOM), originally introduced by Kohonen (1982), to study dialogue act recognition on the basis of the words that occur in the utterance.

The SOM is an unsupervised artificial neural network model where the input data for the model is described in terms of vectors, each of which typically consists of components representing an elementary feature of the data item, expressed as a numeric value. However, our model varies slightly from this basic model. The output is a similarity-based map of the data items, similarity being defined as proximity of items in the vector space. SOM has been applied to various pattern recognition tasks, and an overview of SOM applications in NLP is given in Honkela (2000).

The self-organizing maps differ from the supervised learning methods in that no external teacher is needed in the learning phase. Restrictive pre-categorizations of data can thus be avoided. Unlike alternative statistical methods such as multidimensional scaling and principal component analysis, the SOM approach also offers a platform for continuous upgrading of the data, a requirement native to online services and learning dialogue systems (Jokinen 2000).

Like all data-oriented clustering techniques, SOM requires human analysis and interpretation, and as such the maps are best regarded as a means to visualize meaningful relations among the data items. In dialogue systems, however, the complexity of the task requires higher-level symbolic categorization and structured processing, as well as some criteria of the desired output: the system should classify its inputs into responses that are appropriate with respect to the underlying task. Although hierarchical natural language processing systems can be built on distributed representation and neural networks, we intend to use minimally analysed information and rather than attempt considerable knowledge engineering. Furthermore, in order to evaluate system performance, and to compare it with other similar systems, it is necessary to have quantitative measures of the classification accuracy. We have thus decided to use the LVQ-algorithm, a classification method that learns to map input vectors via a set of codebook vectors which are placed within each class type so that the class-similarity can be approximated by the nearest neighbour -algorithm.

5 Experiments

We have experimented with two classification methods:

1) bag of words with IDF-weights
2) LVQ a) with all words
   b) with content words only.

For both methods the corpus is preprocessed in a similar way: spoken language and slang expressions are normalized, and inflected word forms are converted to their base form by the two-level morphological analyser (Koskenniemi 1983). Ambiguous word forms are solved heuristically by favouring the shortest and simplest analysis. The five morphological features (Table 2) are retained and treated as words. The part-of-speech category is used to pick up nouns, verbs and adjectives for the LVQ processing to test whether there is a difference in classification accuracy if only the content words are used compared to all words.

Preprocessing also includes identification of the most distinctive words for the dialogue acts by calculating their inverse document frequency (IDF), commonly used in document retrieval research to give weight on characteristic words (Sparck-Jones 1972). Term weight *w* for a word is calculated with the formula:

(1) $w = freq \times \log (N / n)$

where *freq* is the frequency of the word in a particular class, *N* is the number of classes, and *n* is the number of classes that the word occurs in. Only the words whose base forms occurred in the corpus more than once and less than 201 times were taken into account.

5.1 Bag of Words

This method corresponds to taking the words that occur in the utterance (bag-of-words) and weighting them according to their significance for each dialogue act type.

Each word is associated with a word vector of length 16, having as component values the term weights associated with each dialogue act class.

Due to the small number of possible classes (N=16), the term weight used inverse document ratio rather than the IDF logarithm, in order to give more emphasis on the IDF.

(2) wordvector for $word_i$ and dacts $dact_j$:

$$w_i = < w_{ij}, w_{ij}, \ldots w_{ij} >$$
$$j = 1 \ldots 16$$

The utterances are processed word by word, and the utterance vector of length 16 is produced by multiplying the components of the normalized word vectors.

(3) utterance vector:

$$u_m = < u_{m1}, u_{m2}, \ldots u_{mj} >$$
$$j = 1 \ldots 16$$

$$u_{mj} = \Pi \, w_{ij}$$
$$i = 1 \ldots k$$
$$k = \text{number of words in } u_m$$

The dialogue act is selected as the one whose component has the highest value in the utterance vector. The accuracy is calculated using 10-fold cross-validation over the whole data.

5.2 LVQ

We use the WEBSOM-style (Kaski *et al.* 1998) method by associating each word in the lexicon with a random vector of length 90 (which has been shown empirically to be a sufficient length, Honkela 1997). Each utterance is treated as one document, and the utterance vectors are formed by summing all the random word vectors for the words that occur in the utterance, weighted by their IDF values with respect to the given Dact class. We created two sets of utterance vectors: Set1 has only the content words, Set2 all the words.

The utterance vectors are then given to the LVQ classification algorithm in LVQ-PAK [2]. A number of 550 codebook vectors is used with the optimized-learning-rate version of the LVQ1 algorithm (Kohonen *et al.* 1996). The LVQ-method requires that care be taken to place the codebook vectors properly in the space so that the class borders follow clear decision lines. The nearest-neighbour rule is used in LVQ-PAK.

6 Results

As a baseline accuracy, we use the accuracy of selecting the most frequent dialogue act, *statement*, i.e. 24%.

[2] http://www.cis.hut.fi/research/som_lvq_pak.shtml

6.1 Bag of Words

The simple bag of words -method achieved an average accuracy of 62%, but varies hugely with different dialogue acts. Recognition of *statements* and meta-communication acts like *thanking*, *opening*, *ending* and *acknowledgement* is excellent, but e.g. only one fourth of *questions* is correctly recognized, and some Dacts are not found at all. It is interesting to note, however, that the 'correct' dialogue act is often the closest competing alternative.

Dialogue Act	Acc. %
Opening	76,97
Sigh	0,00
Call-to-continue	0,00
Thanking	97,60
Correction	0,00
Question	24,61
Ending	94,86
Wait	25,00
Repetition	2,00
Request-to-repeat	50,00
Completion	0,00
Confirmation	0,00
Check	0,77
Answer	2,52
Statement	95,25
Acknowledgement	83,24
Average	62,00

Table 3. Bag of words accuracy.

We assume that the errors are partly due to the small number of examples in the corpus, and partly because the examples are short and the words occurring in them among the common ones: significant word distributions are too similar. On the other hand, the term weighting method tends to favour Dact classes that are larger with respect to the number of words occurring in them, since the term frequencies will tend to be larger for those classes. Looking at the confusion table in Table 4 (which only lists the most common errors), we see that the incorrectly assigned Dact is indeed usually the largest Dact *statement*, while only a few *statements* are confused with the other Dacts. To avoid this bias, the term frequency should thus be normalized by the number of words in the Dact class in question.

Tagged	Assigned	Number
Opening	Ending	27
Call_to_continue	Acknowledgement	35
Correction	Statement	17
Question	Statement	125
Wait	Statement	16
Repetition	Answer	15
Repetition	Statement	55
Confirmation	Statement	91
Confirmation	Acknowledgement	68
Check	Statement	104
Answer	Statement	160
Acknowledgement	Statement	53

Table 4. Partial confusion table.

6.2 LVQ

The results of the LVQ classification are given in Table 5. Set2, i.e. LVQ with all categories, performed significantly better than Set1, LVQ with only the content words. The reason may be the larger training material available, but the result also supports the intuitive view that the content words (nouns, verbs, adjectives) are not alone enough to distinguish among dialogue acts: while they play a big role in topic recognition, the speakers' intentions are also encoded in function words, which thus are important in dialogue act recognition. It should be noticed that variation of the accuracy rates is small among Dact types, and all the rates are above the average accuracy of the BAG-model, except for *sigh* and *completion* which are rather peculiar categories in the first place.

6.3 SOM

The number and type of dialogue acts is not fixed but depends on the theoretical premises as well as on the type of the dialogues being studied (e.g. information-seeking, negotiation, argumentation). Although there are on-going activities to standardize the set of dialogue acts and their definitions (e.g. the DRI-initiative, Carletta et al. 1997), practical dialogue systems make use of their own classifications designed for the application in hand. For instance, our Dact classes are based on the characteristics of the corpus although they take the DRI-standards into account. Thus we also wanted to explore if our classes, based on human linguistic expertise, can find support in the intrinsic categories learnt from the data automatically.

	Set1	Set2
Dialgoue Act	Acc.%	Acc.%
Statement	95,03	96,02
Acknowledgement	91,89	94,62
Question	83,81	95,73
Answer	73,48	92,04
Confirmation	66,67	98,76
Opening	94,44	98,09
Check	73,91	89,28
Repetition	67,65	76,71
Thanking	95,58	94,55
Ending		100
Call_to_continue	33,33	93,75
Correction	78,57	100
Completion	0	16,67
Request_to_repeat	70	72,73
Wait	91,3	95,45
Sigh	0	0
Total	86,55	94,23

Table 5. LVQ Accuracy.

We produced a SOM for the same input space, using the standard algorithm found in SOM-PAK, with a hexagonal topology and a bubble neighbourhood. The ordering phase lasted 10000 steps with the learning rate 0.4 and the neighbouring radius set to 12. The tuning phase lasted 100000 steps with the learning rate and neighbouring radius set to 0.1 and 3, respectively. Although we experimented with both parameters, the final maps did not vary much. After training, the entire input data set is mapped one by one onto the map with the winning node being labelled according to Dact type with which the input was tagged. These hits on each node are then tallied up and the final label on each node represents the number of hits for each Dact type that the node received. A typical map using the parameter settings above and 24x12 units is presented in Appendix 1. As can be seen, clear dialogue act clusters emerged. Most clusters correspond to pre-established Dact categories such as the metacommunication Dacts *thanking, opening* and *ending,* as well as *statement* and *question*, and thus seem to support the intuitive classification. However, it is also interesting to note the formation of clusters which represent a combination of several different categories: the word features used in the utterance vectors are not discriminative enough.

7 Conclusion and Future Work

We have experimented with different methods to classify dialogue acts. The results show that the LVQ-method, using all the words occurring in the utterances, results in excellent accuracy 94%, and outperforms the LVQ using only the content words (accuracy 86%) as well as the "naïve" BAG-method (accuracy 62%). LVQ seems to compare favorably also with other neural and machine learning techniques (cf. e.g. Wermter *et al*. 2000, Möller 1996), and it would be interesting to conduct systematic tests on these.

Using the same minimally analysed information as an input, we also found that the SOM forms clusters of the input which can be interpreted as a classification of dialogue acts, thus providing support for the intrinsic nature of our Dact categories.

Future research will aim to improve the LVQ classification by taking semantic classification of the words into account. The "bag-of-words" approach overlooks semantic similarities between words that occur in similar syntactic positions, whereas in natural language processing such semantic categories play an important role as an equivalence class for words with different realizations.

Another line of research includes experiments on utterances with only out-of-vocabulary words encoded as zero-vectors. Prediction of dialogue acts for utterances that contain no words from the known vocabulary is a practical problem for dialogue systems with speech input, and requires that the position of the utterance in the dialogue context be taken into account.

Acknowledgements

The research was carried out in the context of the USIX Interact project, a collaboration project between four Finnish universities, and funded by the National Technology Agency, the leading IT companies ICL Invia oyj, Sonera oyj, Lingsoft oy, and Gurusoft oy, as well as the Finnish Association for the Deaf and the Arla Institute.

References

Alexandersson, J., R. Engel, M. Kipp, S. Koch, U. Kussner, N. Reithinger, and M. Stede (2000). Modeling Negotiation Dialogs. In W. Wahlster (Ed.) *Verbmobil: Foundations of Speech-to-Speech Translation* Springer Verlag, pp. 441-451.

Allen, J., D. Byron, M. Dzikovska, G. Ferguson, L. Galescu, and A. Stent (2000). An architecture for a generic dialog shell. *Natural Language Engineering* 6(3): 1-16.

Allwood, J., D. Traum and K. Jokinen (2000). Cooperation, Dialogue, and Ethics. In *International Journal for Human-Computer Studies* 53: 871-914.

Carletta, J., N. Dahlbäck, N. Reithinger and M. Walker (eds.) 1997. *Standards for Dialogue Coding in Natural Language Processing*. Dagstuhl-Seminar Report 167

Honkela, T. (1997). *Self- Organizing Maps in Natural Language Processing.* PhD Thesis, Helsinki University of Technology, Dept of Computer Science and Engineering.

Honkela, T. (2000). Self-Organizing Maps in Symbol Processing. In S.Wermter and R. Sun (eds.) *Hybrid neural Systems.* Springer Verlag, Berlin, pp. 348-362.

Jokinen, K. (1996). Goal Formulation based on communicative principles. *Procs of the 16th COLING*, pp. 598- 603.

Jokinen, K. (2000). Learning Dialogue Systems. *Procs. of the LREC workshop From Spoken Dialogue to Full Natural Interactive Dialogue*, pp.13-17.

Jokinen, K. (2001) The Interact project. *Elsnews* 10.2:10 http://www.elsenet.org

Jokinen, K. and G. Wilcock (2001). Confidence-based Adaptivity in Response Generation for a Spoken Dialogue System. *Procs. of the 2nd SIGDial Workshop.* Aalborg, Denmark. pp. 80-89.

Jönsson A. (1997). A model for habitable and efficient dialogue management for natural language interaction. *Natural Language Engineering* 3(2/3): 103-122.

Kaski, S., T. Honkela, K. Lagus, and T. Kohonen (1998). WEBSOM self-organizing maps of document collections. *Neurocomputing*, 21: 101-117.

Kohonen, T. (1982). *Self-organized formation of topologically correct feature maps*, Biological Cybernetics 43:59-69.

Kohonen, T., J. Hynninen, J.Kangas, J. Laaksonen, and K. Torkkola (1996). *LVQ_ PAK: The Learning Vector Quantization Program Packag.* Report A30, Laboratory of Computer and Information Science, Helsinki University of Technology.

Koskenniemi, K. (1983). *Two-Level Morphology: A General Computational Model for Word-Form Recognition and Production*. Publications of the Dept. of General Linguistics, University of Helsinki.

Larsson, S., A. Berman, J. Bos, L. Grönqvist, P. Ljunglöf and D. Traum (2000). TrindiKit 2.0 Manual.

Litman, D., M. Kearns, S. Singh and M. Walker (2000). Automatic Optimization of Dialogue Management. In *Procs. of the 18th COLING*, pp. 502-508.

Miikkulainen, R. (1993). *Sub- symbolic Natural Language Processing: An Integrated Model of Scripts, Lexicon, and Memory*. MIT Press, Cambridge.

Möller, J.-U. (1996). Using DIAMOLE for unsupervised learning of domain specific dialogue acts from spontaneous language. Technical Report FBI-HH-B-191/96, University of Hamburg.

Nagata, M. and T.Morimoto (1994).An information-theoretic model of discourse for next utterance type prediction. In *Transactions of Information Processing Society of Japan* 35(6):1050-1061.

Reithinger, N. and E. Maier (1995). Utilizing statistical dialogue act processing in Verbmobil. In *Procs. of the 33rd Annual Meeting of ACL*, pp.116-121.

Sadek, D., P. Bretier and F. Panaget (1997). ARTIMIS: Natural dialogue meets rational agency. In *Procs of IJCAI-97*, pp. 1030-1035.

Sparck Jones, K. (1972). A Statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* 28(1): 11-21.

Turunen, M. and J. Hakulinen (2000). Jaspis – a framework for multilingual adaptive speech applications. In *Procs. of the 6th International Conference on Spoken Language Processing*, Beijing.

Wermter, S., G. Arevian and C. Panchev (2000). Towards Hybrid Neural Learning Internet Agents. In S. Wermter and R. Sun (eds.) *Hybrid neural Systems*. Springer Verlag, Berlin, pp. 158-174.

Wermter, S. and R. Sun (2000). An Overview of Hybrid Neural Systems. In S. Wermter and R. Sun (eds.) *Hybrid neural Systems*. Springer Verlag, Berlin, pp.1-13.

Wermter, S. and V. Weber. SCREEN: Learning a flat syntactic and semantic spoken language analysis using artificial neural networks. *Journal of Artificial Intelligence Research* 6(1): 35-85.

Appendix 1

Self-Organizing Map of the Input Space