The Neural Path to Dialogue Acts

Michael Kipp[1]

Abstract. This paper presents a neural network approach to the problem of finding the *dialogue act* for a given utterance. So far only symbolic, decision tree and statistical approaches were utilized to deal with a corpus as large as the VERBMOBIL corpus. We propose solutions to the questions of representing speech, network architecture and training in this context. We argue that, when using neural networks, a task like this can only be solved in a modular approach where training data is split and processed by different components of a larger network. Special care must be taken in constructing a feeding mechanism that avoids oscillatory behaviour due to the heterogeneous data.

We were successful in constructing a modular neural network that yielded interesting time-sensitive properties as well as recognition rates superior to most other methods. A first attempt at devising a hybrid system got very close to the best results of this field which suggests further enhancement in future architectures.

1 Introduction

Dialogue acts are a widely used means of representing the intention of a speaker in interactive systems (cf. [2]). Being derived from *speech act theory* [14] which interpreted utterances as actions, assigning each dialogue contribution an *illocutionary force*, computer scientists soon realized the potential of this idea, exploiting these acts as plan operators to model the intentional structure of a dialogue. Examples for dialogue acts are SUGGEST, ACCEPT or THANK (table 3 lists all acts used in this work). Obvious applications are counselling, tutoring and translation systems. VERBMOBIL [15] is one of the latter and relies in various respects on dialogue acts [7][1]. The most important one is that of the *translation objective* [12] defining the central aspect of an utterance that has to be carried over to the target language (the propositional content should be added to make the information complete, e.g. for a suggestion it is not enough to know that it is a SUGGEST dialogue act but also what the suggestion was about).

Since dialogue acts model different dimensions of communication at once, overlaps occur (e.g. countersuggestions could be viewed as being a REJECT and a SUGGEST at the same time). Therefore, in VERBMOBIL each utterance is labelled with multiple dialogue acts. In this work, however, multiple acts are, for reasons of simplicity and comparability, ignored, i.e. for every utterance there is one annotated dialogue act (for recent research on the issue of dialogue acts cf. [3]).

So how do we find out the dialogue act for a given utterance? Several suggestions arose, mainly using symbolic means like hand-coded rules [12] or statistical means [10][9][13]. The latter has been the most successful method so far although it has not been able to improve much beyond results that could be easily obtained with simple bigrams [10]. This seems to be due to problems with the training corpus size which usually does not have enough samples to provide even 3-gram models with sufficient data, let alone higher n-gram models.

Hence, the idea to employ neural networks that might prove more suitable, or at a later stage, might contribute in a hybrid system to solve the task better. We focus our attention on recurrent networks like Elman and Jordan nets that have time-sensitive properties [4][8]. These networks introduce a strongly restricted form of recurrency that still allows controlled processing and the utilization of the standard backpropagation algorithm. Figure 1 shows an Elman net which is basically a simple Feedforward net with one hidden layer and a special so-called context layer where the values of the hidden neurons are shifted to after each step of processing.

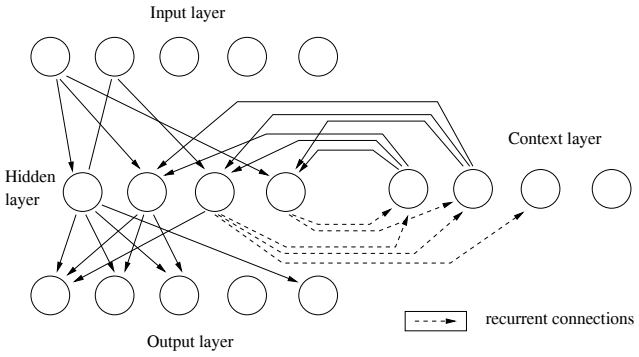

Figure 1. architecture of a simple Elman network with one hidden layer and one context layer

The many promising properties of neural networks like robustness, parallel and incremental processing, easy adaptability to new domains by re-training (since there are no hand-coded rules) led to this work. We tackle the problem by providing working solutions to a sequence of three subproblems:

1. Finding a suitable representation for the input data (transcribed spontaneous natural language utterances)
2. Devising an overall architecture of a modular network
3. Preparing the training data according to the needs of the task and feeding the network in a suitable way

In the further course of the paper we will present our solutions, evaluate them with respect to related work and elaborate on possible future improvements of this project.

2 Representation

An artificial neural network processes information by modifying the numerical values of its neurons that usually lie in an interval of $[0, 1]$.

[1] DFKI GmbH Saarbrücken, Germany, kipp@dfki.uni-sb.de

ECAI 98. *13th European Conference on Artificial Intelligence*
Edited by Henri Prade
Published in 1998 by John Wiley & Sons, Ltd.

It retrieves its input from a series of designated input neurons. The number of these neurons is constant. So how do we feed a network accepting input of constant length with utterances of variable (and possibly very large) size? Different solutions exist:

- Entering an utterance as a whole, leading to large networks which could still be too small for certain utterances
- Using a sliding window mechanism where a part of the utterance of fixed length (e.g. three words) is fed to the network moving from the first part to the last
- Exploiting the time-sensitive properties of recurrent networks like the Elman or Jordan networks [4][8] which use a context layer of neurons as a kind of memory. Utterances can be processed word by word or in bigger units.

The first possibility was ruled out due to restrictions in size which are necessary to limit training time. Instead a combination of the two last techniques led to optimal results, i.e. taking a partially recurrent neural net and feeding it with the contents of a sliding window (a sliding window of size one becomes the third alternative). Thereby, context information enters the network at two stages. First, through the syntactic properties of the input vector and second, through the buffer memory of the partially recurrent network.

This leaves us with the problem of representing a single word as a vector. The optimal solution for neural processing would be the so-called *1-of-C representation* where each possible item (here: word) gets its own input neuron. With a lexicon of about 3800 word tokens it proved impossible to train the resulting networks within sensible time limits. Therefore, some form of compression was necessary. On top of this we aimed at including information about the syntactic category, namely the *part-of-speech* (POS) of a word, since

1. in some cases the usage of parts-of-speech could reveal a more general template-like structure of dialogue act utterances:

   How about <ADJECTIVE> <NOUN> ? `(suggest)`

   That <VERB> okay. `(accept)`

2. in case of unknown words having part-of-speech information could help the system work with minimal derogation

The final design comprised a vector with one segment for each part-of-speech category (we devised 15 such categories). A <NOUN> would be represented exclusively within the special NOUN segment of this vector. Each POS segment could therefore use its own representation for all the words belonging to this POS. We chose each representation according to the importance of differentiating within the respective POS category. The category <PRON> (containing words like `where, what, how, ...`) for example is certainly more worthwhile to look at closely than the category <CARD> (containing cardinal numbers).

Words within a POS of high importance (relative to the task) are thus represented by assigning to each possible word one distinct component (1-of-C representation). Within a POS of a medium importance words were represented by the binary of their position in the word list of their category. Finally, the words of low importance (i.e. where only the category itself is interesting) were grouped to a single component of the vector, i.e. in case of a cardinal number the system would only get to know that there was a cardinal number but not which one. Figure 2 shows a sample input vector of the word token "well" (adverbs were represented as binaries).

The resulting representation made use of 15 POS categories and yielded an input vector with 216 components.

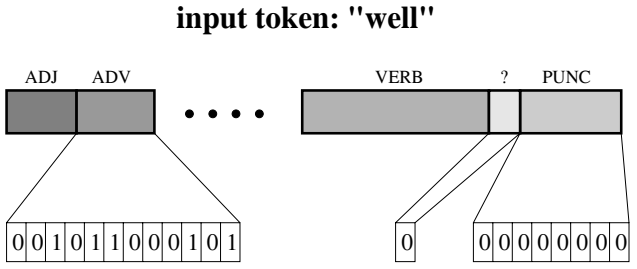

Figure 2. input vector representation of the adverb "well"

3 Architecture

Design and training of the networks was done with the Stuttgart Neural Network Simulator (SNNS) which allows to construct and complile different kinds of neural networks [18].

For the overall architecture we decided on a modular approach after some discouraging experiments with a monolithic network. A natural way of splitting the task of dialogue act annotation was to consider the partial task of detecting one special dialogue act in an utterance. Accordingly, we trained specialized networks for each dialogue act, giving each net one YES and one NO neuron as output units. The resulting YES/NO outputs are not probabilities nor are they in any way comparable amongst each other. Still one could simply pick the net with the highest (YES − NO) value and declare the associated dialogue act the best guess for the current utterance. A better solution is to modify each network output with weights optimized on the training corpus. As a neural network does exactly that, i.e. weighting and recombining values, we constructed another neural net taking the output of the modular nets for a whole utterance[2] and learning to modify the outputs in such a way as to correct the missing comparability amongst the modular nets. This selector network improved recognition rates by about 10%. The general architecture is shown in figure 3.

The selector network did not have to have time-sensitive properties but using an Elman/Jordan network yielded the best results. This indicates that the more global context of preceding dialogue acts was exploited to improve recognition rates. The importance of taking the information of preceding dialogue acts into account has been shown in [10].

4 Training

Training was done on the VERBMOBIL corpus which consisted of 467 German dialogues in the domain of appointment scheduling. These dialogues had been recorded at different German universities and institutes, manually transcribed and manually annotated with dialogue acts according to the guidelines in [7].

For the training of the 18 modular networks we had to determine (a) how to organize the set of data (e.g. into smaller packages), (b) how to tune learning parameters, and (c) when to stop training.

4.1 Training in Packages

The data for the modular network representing dialogue act d consisted of all utterances in the training corpus with dialogue act d. We

[2] the output of a single modular network d for an utterance $u = (w_0, \ldots, w_n)$ with output functions yes_d and no_d is computed by the formula $\frac{1}{n}\sum_{k=0}^{n}(\text{yes}_d - \text{no}_d)$.

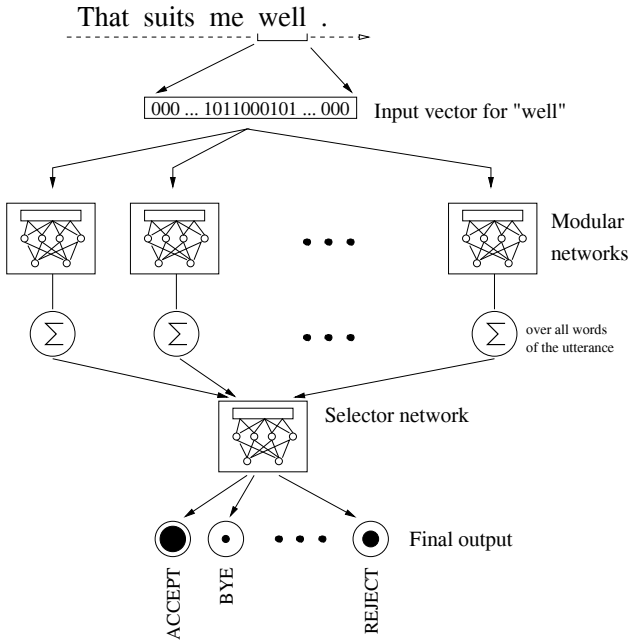

Figure 3. overall architecture showing the modular networks - one for each dialogue act - plus the selector network

called these utterances the *positive* patterns, as opposed to the *negative* patterns which was all the rest of the training data.

The capacity of a neural network is limited by the size of its hidden layer which is in turn limited by computational resources and architectural considerations. Feeding all available data to such a limited network where the hidden layer contained 30–90 neurons resulted in poor performance. We therefore divided the training corpus in packages where training ran for a certain time whereafter we would present the next package. Each package contained all positive utterances plus an equal number of negative utterances, all encoded in aforementioned representation. These packages were ordered according to a simple complexity measure following Elman's advise to train simple patterns first [5]. On top of this we included a training supervision mechanism that would measure progress of the training and switch packages in case of bad progress or stretch the training length in case of good progress where progress was measured with the *middle squared error* (MSE). Figure 5 shows the training error development with this kind of training whereas figure 4 depicts the original curve of the conventional feeding of all patterns subsequentially for a number of epochs.

4.2 Tuning Parameters

The training parameters for the modified backpropagation learning algorithm were tuned according to standard literature [6][18]. Backpropagation works as a series of weight modifications leading an error function E to a minimum. The *learning parameter* η determines the quantity of each modification and is picked from an interval of $(0, 1)$.

For the modular networks we chose a small learning rate of $\eta = 0.05$ because of the high amount of relatively heterogeneous data that we did not want to make the net converge too quickly to a local minimum. For the selector network $\eta = 0.2$ proved suitable. Since we made use of the *backpropagation with momentum* version to make

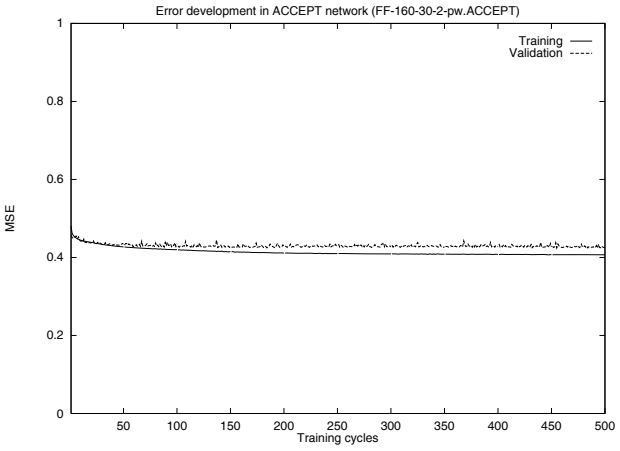

Figure 4. training curve measuring MSE over number of epochs (subsequent pattern feeding)

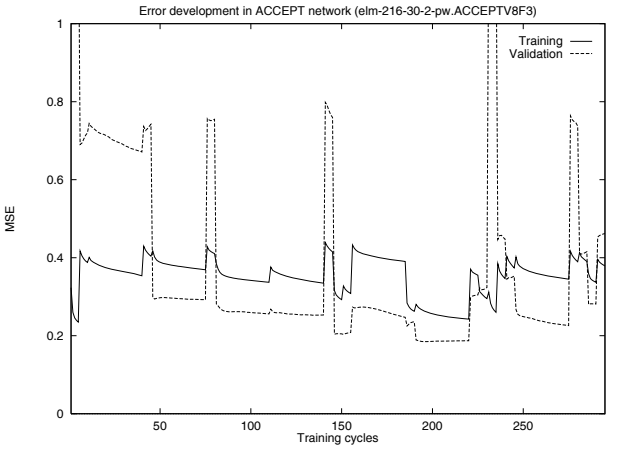

Figure 5. training curve measuring MSE over number of epochs (adaptive feeding in packages)

convergation faster (cf. [6] for details) we had to tune the momentum term μ, setting it to 0.5. In all changing these parameters a little did not show any significant effect on the performance of the nets.

4.3 When to Stop

The last and one of the most decisive problems is that of training duration since a neural network quickly overadapts to training data compromising its performance on unseen data to an unacceptable degree (also called *overfitting*). The usual step taken is to split the training set into a training corpus and a validation corpus. The latter is used to test the network's performance on data not occurring in the training patterns which should give the system an idea of the networks ability to generalize. Figures 4 and 5 show the results on the validation corpus with a dashed line. Since the MSE does not give much information about the quality of the annotation process which happens on a higher level we had to define another measure to select the optimal training duration. This measure combined `recall` and `precision` of the annotations of a single net (YES/NO annotations) to determine the best modular net in each of the 18 training runs. The

exact formula was

$$\text{quality(d,k)} = \alpha \times \text{recall(d,k)} + (1 - \alpha) \times \text{precision(d,k)}$$

where d denotes the network and k the number of training epochs. In most of our experiments $\alpha = 0.4$ worked best.

5 Results on Context

Examining the behaviour of the best network on test sentences brought about interesting results concerning the network's ability to react to context. Although in our final architecture the net only retrieves one word at a time, it is able to distinguish equal words relative to context like in:[3]

Text	JA DAS WÄRE GANZ GUT *yes, that would be great*	ACCEPT
Netoutput	ACCEPT 0.59699	
Text	WIR MÜSSEN JA MAL WIEDER DEN TERMIN FÜR (...) FESTSETZEN *well, we do have to fix the date for (...) once more*	INIT
Netoutput	ACCEPT -0.10364	
Text	SCHÖNEN DANK AUCH *thank you then*	THANK
Netoutput	ACCEPT -0.21865 THANK 0.19883	
Text	JA DER NEUNZEHNTE PAßT MIR AUCH SEHR GUT *yes, the nineteenth suits me well, too*	ACCEPT
Netoutput	ACCEPT 0.36413 THANK -0.96152	

Other examples show that the networks even made use of information outside the current utterance's borders, i.e. of the previous utterance, like in the utterance

JA GUT

which has a similar meaning and ambiguity as the English *okay* that can be used to signal agreement (ACCEPT) or to confirm one's interest in the conversation (FEEDBACK). It was correctly classified by the network. These are only some examples. A more systematic analysis of the network's properties is one of our future aims.

The final overall evaluation was conducted on a previously unseen set of data, disjoint with training and validation sets.

6 Evaluation

Our experiments ran on the VERBMOBIL corpus of 467 German appointment scheduling dialogues. It was partitioned into training, validation and test sets as shown in table 1.

[3] the following table shows processed utterances of the test corpus in the middle section (translation in italics). The currently processed word is underlined (a more or less equal word in the translation is underlined for better understanding though the point is lost in all cases by the translation), the activation of the most interesting modular net(s) is given underneath in the form (YES - NO). The correct dialogue act is right of the utterance.

unit	*training*	*validation*	*test*	*total*
dialogues	350	87	30	467
utterances	10766	2903	852	14521

Table 1. partitioning of the dialogue corpus

First experiments with monolithic networks yielded a recall of 45.11% in the best configuration (Feedforward net with 250 hidden units and a sliding window size of 2) which led us to the conclusion that only a modular approach could cope with this kind of data.

The best modular network annotated the test data with a recall of 60.45% after a couple of enhancements in training and adding a selector network as described above. The most important results are summerized in table 2 showing the impact of the respective modifications. A more detailed view on the results is offered in table 3 where precision and recall for each single dialogue act are given.[4]

	without selector	*with selector*
conventional feeding	48.29%	57.84%
package feeding	57.84%	60.45%

Table 2. results of modular networks

dialogue act name	*recall*	*prec.*	*occ.*	*ann.*	*corr.*
THANK	100.0	90.0	9	10	9
GREET	93.55	93.55	31	31	29
INTRODUCE	92.86	100.0	14	13	13
SUGGEST	86.28	60.19	226	324	195
BYE	85.42	89.13	48	46	41
INIT	67.44	65.91	43	44	29
REQUEST_COMMENT	66.67	77.78	21	18	14
REQUEST_SUGGEST	61.54	61.54	26	26	16
ACCEPT	53.7	46.77	108	124	58
FEEDBACK	52.94	52.94	51	51	27
REJECT	52.86	50.0	70	74	37
GIVE_REASON	50.0	55.81	48	43	24
DIGRESS	38.46	50.0	13	10	5
DELIBERATE	33.33	56.52	39	23	13
CONFIRM	14.29	25.0	7	4	1
GARBAGE	5.0	33.33	20	3	1
CLARIFY	3.94	37.5	76	8	3
MOTIVATE	0.0	-.-	2	0	0

Table 3. results of best configuration

The result of the network compares pretty well in the field of automated dialogue act annotation where different solutions have been offered in the past. Symbolically operating approaches like the system FLEX [12] yielded high results but always tested their handcoded rules on the data the rules were extracted from, a methodology

[4] the three rightmost columns give figures for the number of occurances (occ), the number of times a dialogue act was annotated (ann) and the number of times a dialogue act was correctly annotated (corr).

that disqualifies the results from comparison. Another project utilized decision or classification trees [9] obtaining a top recall value of 46%, therefore proving inferior to our approach. Recently, there have been attempts to devise symbolic rules using a Monte Carlo algorithm. This so-called *transformation-based learning* has reached very good results close to statistical ones (cf. [11]).

On the other hand, statistical n-gram methods are still unbeaten [10][9][16]. The work of [10] reached a recall of 67.53% on exactly the same data as this project which seems a long way ahead. One has to consider, though, that simple bigrams already give a recall of 65.73%, leaving an improvement of the much more elaborate systems by about 2%. Therefore, it appears as though statistical approaches have already found their limits whereas neural networks yield a huge number of possibilities still to be explored. In a hybrid version of this work – more closely described in the next section – we were able to obtain a recall of 66.31% on unseen data which is almost as good as the best result in this field.

7 Hybrid Steps

One first approach towards combining statistics and neural networks was done by replacing the original speech representation by a vector that basically consists of statistical information. This representation was developed in [17], another neural network project that used a monolithic design on a small corpus. Each word w is represented by a vector of length 18, one component for each dialogue act. Component d retrieves the estimated probability $P(d|w)$, i.e. the probability that word w indicates a dialogue act annotation of d. A word's representation is therefore its probability distribution over all dialogue acts.

Replacing our representation by this statistical distribution (input being restricted to one word per go), otherwise using exactly the same architecture and parameters, we achieved a recall value of 66.31% on the test set.

Future directions would first of all look into the potential of widening the input window to two words where we would use the probabilities of $P(d|w_1, w_2)$.

8 Conclusion and Future Work

In this paper we have outlined the design and capabilities of a modular neural network that automatically annotates utterances with dialogue acts. Different design decisions were presented and justified. Representation of speech was done by multiple section vectors including parts-of-speech information. A modular architecture was proposed to cope with the large amount of data. The architecture was based on the dialogue acts themselves, assigning one neural net to each act and combining/interpreting the single results by a selector network. Training was managed in packages and the package training duration was made dependent on the network error rate. The resulting network was analyzed and compared with related work in the field, proving its superiority over all other approaches but the statistical one, although first experiments with a hybrid system resulted in recall values only 1% worse than those of the n-gram method.

Future research will involve the close examination of the weaknesses and strengths of the statistical and the neural network approach. Then one could cluster dialogue acts to groups and let one method recognize the group and the other deal with specifying the member of the group. Ideally, the clustering should be based on automatically detected correlations.

The design of the neural network itself is certainly open for future improvement. One major drawback of neural networks is the training time which is strongly linked with representation issues (since the representation determines the size of the network). Furthermore, an analysis of the interior representation of words could give important clues as to how to improve the original representation.

REFERENCES

[1] Jan Alexandersson, Elisabeth Maier, and Norbert Reithinger, `A Robust and Efficient Three-Layered Dialog Component for a Speech-to-Speech Translation System', in *Proceedings of the 7th Conference of the European Chapter of the ACL (EACL-95)*, pp. 188–193, Dublin, Ireland, (1995). also available as Verbmobil Report Nr. 50, DFKI GmbH, Dezember 1994. available in the cmp-lg electronic archive under no. cmp-lg-9502008.

[2] Harry C. Bunt, `Rules for the Interpretation, Evaluation and Generation of Dialogue Acts', in *IPO Annual Progress Report 16*, pp. 99–107, Tilburg University, (1981).

[3] Mark Core and James Allen, `Coding Dialogs with the DAMSL Annotation Scheme', in *AAAI Fall Symposium on Communicative Action in Humans and Machines*, Boston, MA, (1997).

[4] Jeffrey L. Elman, `Finding structure in time', *Cognitive Science*, **14**, 179–211, (1990).

[5] Jeffrey L. Elman, `Distributed representations, simple recurrent networks, and grammatical structure', *Machine Learning*, **7**, 195–225, (1991).

[6] John Hertz, Anders Krogh, and Richard G. Palmer, *Introduction to the Theory of Neural Computation*, Addison-Wesley Publishing Co., 1991.

[7] Susanne Jekat, Alexandra Klein, Elisabeth Maier, Ilona Maleck, Marion Mast, and J. Joachim Quantz, `Dialogue Acts in VERBMOBIL', Verbmobil Report 65, Universität Hamburg, DFKI Saarbrücken, Universität Erlangen, TU Berlin, (1995).

[8] M. I. Jordan, `Serial order: A parallel, distributed processing approach', in *Advances in Connectionist Theory: Speech*, eds., J. L. Elman and D. E. Rumelhart, Hillsdale, (1989). Erlbaum.

[9] M. Mast, E. Nöth, H. Niemann, and E.G. Schukat-Talamazzini, `Automatic Classification of Speech Acts with Semantic Classification Trees and Polygrams', in *International Joint Conference on Artificial Intelligence 95, Workshop "New Approaches to Learning for Natural Language Processing"*, pp. 71–78, Montreal, (1995).

[10] Norbert Reithinger and Martin Klesen, `Dialogue act classification using language models', in *Proceedings of EuroSpeech-97*, pp. 2235–2238, Rhodes, (1997).

[11] Ken Samuel, Sandra Carberry, and K. Vijay-shanker, `Computing dialogue acts from features with transformation-based learning', in *Proceedings of the AAAI 98*, pp. 90–97, (1998).

[12] Birte Schmitz and J. Joachim Quantz, `Dialogue Acts in Automatic Dialogue Interpreting', in *TMI-95*, Leuven, (1995).

[13] E. G. Schukat-Talamazzini, *Automatische Spracherkennung – Grundlagen, statistische Modelle und effiziente Algorithmen*, Künstliche Intelligenz, Vieweg, Braunschweig, 1995.

[14] John R. Searle, *Speech Acts.*, Cambridge/GB: University Press, 1969.

[15] Wolfgang Wahlster, `Verbmobil–Translation of Face-to-Face Dialogs', Technical report, German Research Centre for Artificial Intelligence (DFKI), (1993). In Proceedings of MT Summit IV, Kobe, Japan, 1993.

[16] V. Warnke, S. Harbek, H. Niemann, and E. Nöth, `Topic spotting using subword units', Technical Report 205, F.-A.-Universität Erlangen-Nürnberg, (März 1997).

[17] Stefan Wermter and Matthias Löchel, `Learning dialog act processing', Technical report, University of Hamburg, (1996).

[18] A. Zell, G. Mamier, M. Vogt, N. Mache, R. Hübner, S. Döring, K.-U. Herrmann, T. Soyez, M. Schmatzl, T. Sommer, A. Hatzigeorgiou, D. Posselt, T. Schreiner, B. Kett, G. Clemente, and J. Wieland, `Snns - stuttgart neural network simulator, user manual version 4.1', Technical Report 6/95, University of Stuttgart - IPVR, (1995).