# Artificial Neural Networks for Automatic Knowledge Acquisition in Multiple Real-World Language Domains

Stefan Wermter   Volker Weber
University of Hamburg, Computer Science Department
Vogt-Kölln-Straße 30, D-22527 Hamburg, Germany
{wermter,weber}@informatik.uni-hamburg.de
http://www.informatik.uni-hamburg.de/Arbeitsbereiche/NATS/staff/{wermter,weber}.html

**Abstract.** In this paper we describe a new approach for learning spontaneous language for multiple domains using artificial neural networks. This approach is based on a novel use of flat syntactic and semantic representations, fault-tolerant processing of noisy spontaneous language, and learning of individual domain-dependent subtasks. This approach has been implemented in our parallel and incremental architecture SCREEN (Symbolic Connectionist Robust EnterprisE for Natural language) which we have based on a careful selection and interaction of symbolic modules and artificial neural networks. We present the learned syntactic and semantic categorization and we examine the potential for increasing the portability by focusing on multiple corpora and domains. We claim that the general properties of learning, fault tolerance, and flat representations as implemented in SCREEN have the potential to increase the portability of neural network-based systems for spontaneous language analysis.

## 1 Introduction

Recently, the field of artificial neural networks for language processing has seen a new emphasis on working towards *real-world* speech/language systems. Early work on neural networks had successfully focused on carefully designed *artificially generated* training material, e.g. for various language tasks [8], case role assignment, [4, 6], sequence processing [2, 7], and rule induction [5]. However, in most cases neural network techniques have not yet been examined under hard *real-world* constraints of unrestricted natural language. This is the topic of this paper and we show that artificial neural networks have the potential to deal with real-world spontaneous language input.

In our previous work we used hybrid connectionist techniques for learning a flat understanding of *text* language [13, 14, 15]. After flat learned analysis could be used successfully for relatively well-formed text analysis [13] we moved stepwise to the analysis of less well-formed spontaneous dialog analysis [12, 17]. In this paper we describe new extensive experiments and results of learning a screening flat understanding of spontaneous real-world *dialog* language in multiple domains and we present the learned syntactic and semantic categorization from our system SCREEN.

The general properties which have been incorporated into the SCREEN system are 1) neural network learning, 2) fault tolerance, and 3) a screening flat representation. Learning reduces the amount of knowledge engineering and increases portability. Fault-tolerance is needed for dealing with errors in spontaneous language like pauses, interjections, repairs, and repetitions. Flat representations are needed for supporting portability and the analysis of syntactically or semantically anomalous spontaneous language. We will start by developing the basis for a learned robust flat analysis. Afterwards, we will describe our results for a flat syntactic and semantic analysis for the domain of interactions at a railway station counter using the RTC corpus[1]. Then we will use the same neural network learning techniques and as much of the underlying system concepts as possible for a new different domain of meeting arrangements using the BMC corpus[2]. Typical sentences in the RTC and BMC domains are:

- Yes I need eh a a sleeping car PAUSE from PAUSE Regensburg to Hamburg (RTC corpus)
- Oh yes right but I have from nine to four already a date therefore I would perhaps PAUSE suggest Thursday (BMC corpus)

First, we will give an overview of the used syntactic categories. Since we address the general problem of representing spontaneous language we have to take into consideration the incremental properties of language processing. In

[1] Corpus compiled at the University of Regensburg (FRG) containing travel inquiries.

[2] Corpus compiled at the University of Karlsruhe (FRG) containing meeting arrangements (also called Blaubeuren dialogs).

particular, spontaneous language can contain repairs, repetitions, interjections, pauses, new starts and many other hesitation errors. Therefore, building a deep structural interpretation will be much less successful than for text understanding and analyzing relatively well-formed sentences. Furthermore, we want to start to process language incrementally without waiting until a complete utterance has been finished. Because of the hesitation errors and the incrementality of language we designed a flat representation of spontaneous language at various basic and abstract levels. Table 1 shows the basic syntactic categories as they have been used for the RTC corpus; table 2 shows the abstract syntactic categories. Rather than having arbitrary deep syntactic representations we have a flat basic and an abstract syntactic representation consisting of category vectors with 13 respectively 8 categories for the representation of the syntactic categories of a word.

| Category | Examples | Category | Examples |
|---|---|---|---|
| Noun | train, track | adJective | early, cheap |
| Verb | need, go | Adverb) | very, perhaps |
| pReposition | from, to | Conjunction | and, or |
| pronoUn | I, you | Determiner | the, a |
| nuMeral | one, two | Interjection | eh, oh |
| Participle | taken | Other | particles |
| pause (/) | pause | | |

Table 1: Elements of the basic syntactic category vector

| Category | Examples |
|---|---|
| Verb Group | would like to have |
| Noun Group | a ticket |
| Adverbial Group | later, as early as possible |
| Prepositional Group | to Hamburg, in the morning |
| Conjunction Group | and, either ... or |
| Modus Group | interrogatives, confirmations: when |
| Special Group | additives like politeness: please, then |
| Interjection Group | interjections, pauses: eh, oh |

Table 2: Elements of the abstract syntactic category vector

Similar to the syntactic representations we represent the flat semantic meaning by using basic and abstract semantic categories. Table 3 shows the basic semantic categories as they have been used for the RTC corpus; table 4 shows the abstract semantic categories.

In summary, we have developed a new flat category representation for processing spontaneous language. Since spontaneous language contains many hesitation errors, ungrammatical constructions, restarts etc., a flat representation at different interpretation levels supports the learning of a robust fault-tolerant processing.

| Category | Examples |
|---|---|
| NEED | need, would like |
| MOVE | go, ride |
| STATE | know, exist |
| AUX | can, could |
| SAY | say, ask |
| QUESTion | question words: which, when |
| PHYSical | physical objects: train, wagon |
| ANIMate | animate objects: I, you |
| ABStract | abstract objects: connection, class |
| HERE | time or location state words, preps.: on, in |
| SouRCe | time or location source words, preps.: from |
| DESTination | time or location destination words, preps.: to |
| LOCation | Frankfurt, Hamburg |
| TIME | tomorrow, at 3 o' clock |
| HOW | with, without |
| NEGation | no |
| NILL | words “without” specific semantics, e.g. the |

Table 3: Elements of the basic semantic category vector

| Category | Examples |
|---|---|
| ACTion | action for full verb events: go, need |
| AUX-action | auxiliary action for aux. events: would like |
| AGENT | agent of an action: I |
| OBJect | object of an action: a ticket |
| RECIPient | recipient of an action: to me |
| INSTRument | instr. for an act.: using a high-speed train |
| MANNER | how to ach. an act.: with switching trains |
| TiMe-AT | at what time: in the morning |
| TiMe-FRoM | start time: after 6am |
| TiMe-TO | end time: before 8pm |
| LoCation-AT | at which location: in Frankfurt |
| LoCation-FRoM | start location: from Dortmund |
| LoCation-TO | end location: to Hamburg |
| QUESTion | question phrases: at what time |
| MISC | miscellaneous words, e.g. for politeness |

Table 4: Elements of the abstract semantic category vector

## 2 Overview of SCREEN

So far, we have described the flat syntactic and semantic representations at the basic and abstract levels. We will now show how these flat representations are integrated in our system SCREEN. SCREEN stands for *Symbolic Connectionist Robust EnterprisE for Natural language* and deals with fault-tolerant learning of spontaneous natural language. Figure 1 gives an overview of the six parts in SCREEN as well as a more detailed description of five essential modules in the category part.

In general, the six parts in SCREEN are the speech sequence construction part, the speech evaluation part, the category part, the correction part, the case frame part and the subclause part. The speech sequence construction part is responsible for receiving the word hypotheses from a speech recognizer and for building sentence hypotheses based on these word hypotheses. The speech evaluation part provides an evaluation of the syntactic and semantic plausibility of a sequence. The category part is the part for learning flat syntactic and semantic category representations and our major focus here for examining portability for different domains. The correction part is responsible for detecting pauses, interjections, word errors, and phrase errors in the form of repetitions and repairs as well as for eliminating them from the current parse and the final interpretation. The subclause part and the case frame part are responsible for distinguishing multiple case frames for sentences with several subclauses.

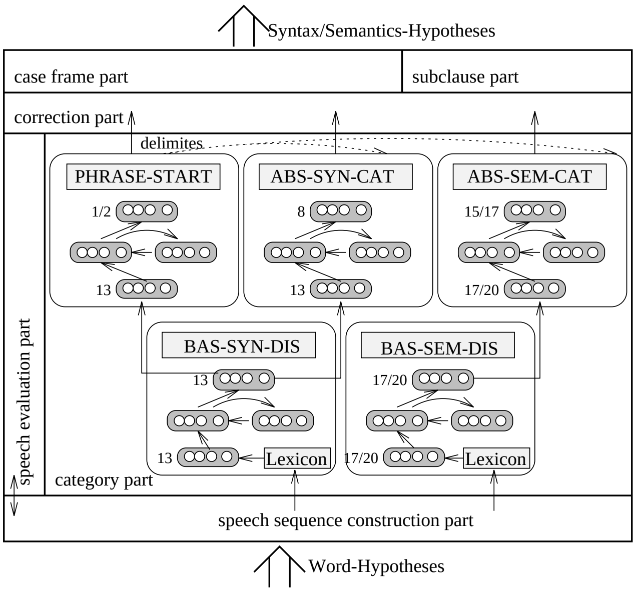


Figure 1: Overview of SCREEN with a focus on the category part

All modules in SCREEN - like the five shown in figure 1 - communicate via symbolic messages and can contain either symbolic representations or trained connectionist networks. The module BAS-SYN-DIS (BAS-SEM-DIS) receives a word as part of a sequence, looks up its basic syntactic (semantic) category representation, and provides a basic syntactic (semantic) disambiguation using the preceding context. For instance, for the word "train" the lexicon can contain the basic syntactic categories "verb" and "noun". However, in a particular sequence like "I need a train" the syntactic category has to be "noun" depending on the previous context. For representing the preceding context we use simple recurrent networks [2]. BAS-SYN-DIS (BAS-SEM-DIS) performs this basic syntactic disambiguation and provides input for the module ABS-SYN-CAT (ABS-SEM-CAT).

ABS-SYN-CAT (ABS-SEM-CAT) is responsible for the abstract syntactic categorization, according to the abstract syntactic (semantic) categories shown in table 2 (4). The modules for abstract categorization receive a representation of a disambiguated word in a particular sequence and provide an abstract syntactic and semantic category representation. For instance, in a simple sentence like "we meet in the morning", "we" would be a noun group, "meet" a verb group, and "in the morning" a prepositional group. Finally, the module PHRASE-START receives the disambiguated basic category representation of a word and supplies hypotheses about the phrase boundaries of a sentence since ABS-SYN-CAT (ABS-SEM-CAT) provides only a categorization for each word. We will come back to a more detailed analysis of the functionality in section 5 after we have described the overall network performance.

The architecture of SCREEN is parallel interleaved and incremental. After a word hypothesis has entered the system, all basic categorization modules for this word hypothesis will run in parallel. At the same time, the abstract categorization modules of the previous word hypothesis of the same sequence run in parallel, since these abstract categorization modules need the output of the basic categorization first. Furthermore, error detection and correction is done in the same parallel interleaved manner as early as possible based on basic and abstract categorization.

# 3 Learning in Plausibility Networks

In this section we summarize briefly the learning rule in recurrent plausibility networks [14]. Recurrent plausibility networks provide a general framework where feedforward networks and simple recurrent networks are special cases of more general recurrent plausibility networks.

Let $L_i(t)$ denote the set of indices of the $ith$ layer at time $t$ with $i \in \{0, \cdots, n\}, L_0(t) = Inputlayer(t)$, and $L_n(t) = Outputlayer(t)$. Then, the input to a unit $j$ is given as:

$$net_j(t) = \sum_z \sum_i w_{ij} y_i(t-z) \qquad (1)$$

for $z \in \{0, \cdots, t_x\}$, $t_x$ is the maximum time step of context layers CL for $L_x$, unit $j \in L_x$ and $x > 0$. Furthermore, $w_{ij}$ is the weight from unit $i$ to unit $j$, $y_i(t)$ is the current computed output value of unit $i$ at time $t$. Finally let $d_{pj}$ denote the desired output value of a unit at the output layer, $\delta_k(t)$ the computed error at hidden units, function $f$ is a semilinear function, that is, the function $f$ is nondecreasing and differentiable. Then, the update rule for a general recurrent plausibility network can be specified as:

$$\Delta w_{ij}(t) \equiv \begin{cases} (d_j(t) - y_j(t))\; f'_j(net_j(t))\; y_i(t) \\ \quad \text{if } i \in L_{n-1}(t), j \in L_n(t) \\ \\ (\sum_k \delta_k(t)\; w_{jk})\; f'_j(net_j(t))\; y^*_i(t) \\ \quad \text{otherwise} \end{cases} \qquad (2)$$

$$y_i^*(t) = \begin{cases} y_i(t) & \text{if } i \in L_{x-1}(t) \\ y_i(t-1) & \text{if } i \in CL_{x-1}(t-1) \\ \vdots & \vdots \\ y_i(t-t_x) & \text{if } i \in CL_{x-1}(t-t_x) \end{cases} \quad (3)$$

This learning rule for plausibility networks applies to hidden layers that can have an arbitrary but fixed number of distributed recurrent connections. This way, the internal dynamic states of a plausibility network over time can be used to introduce incremental context and sequentiality in a general manner. In general, this learning rule provides a common framework for feedforward networks, simple recurrent networks and generalized plausibility networks with an arbitrary number of hidden and context layers. In this paper, especially later in section 5 we focus on the special case of simple recurrent networks.

# 4 Portability experiments: Learning flat representations in two different domains

In this section we describe the training and generalization results for two domains. Our first domain was the RTC corpus which contained 175 sentences from the domain of interactions at a railway station counter. The words of these sentences were labeled with their basic syntactic, abstract syntactic, basic semantic, and abstract semantic categories as well as with the phrase starts. For this domain we used the categories as they were described in tables 1, 2, 3, and 4. Then five simple recurrent networks were trained for the five subtasks using supervised learning [9]. The total number of words for the 175 sentences was 3683. Our initial training set contained 1/3, the test set 2/3 of the sentences. Table 5 shows the results of the different modules. As we can see, some modules, especially the syntactic modules like BAS-SYN-DIS with 99% and 93% and ABS-SYN-CAT with 91% and 85% perform quite well. A word categorization is counted as correct, if the generated network category agrees with the desired category.

After we had reached this level of performance on the RTC corpus we wanted to test our concepts of flat representations, learning, and fault tolerance for a different domain in order to evaluate the portability of the concepts and its implementation in the SCREEN architecture. Therefore we chose the BMC corpus which contains 184 sentences from the domain of meeting arrangements between business partners. We used 1/3 of these sentences for training and 2/3 for testing. The total number of words for the 184 sentences was 2355.

An important question was how much of the underlying category representations we had to change for the new corpus, in order to test portability and generality of the SCREEN concepts and architecture. Since the syntactic categories (but not the constructions) are relatively domain-independent, the same basic and abstract syntactic representation could be used for the BMC corpus as for the RTC corpus. However, for the semantic categories we had to make several changes, due to the different domains. While the railway domain RTC contained the events “need”, “state”, the meeting domain BMC required the events “suggest”, “meet”, “select”, “is”, “have”. The only other necessary changes were the addition of a category “yes” for positive answers in addition to the existing “no” for negative answers and the removal of the “how” category in the BMC corpus. The other categories for objects, time and locations could be used as before. Therefore we had 20 rather than 17 basic semantic categories. For the abstract semantic categories we added two categories in the BMC corpus, “confirm” and “negation”, in order to reflect the many positive or negative reactions in the BMC corpus. Therefore, we had 17 rather than 15 abstract semantic categories. Of course, as usual with semantic representations, it might be argued that a slightly different set could have been used. However, given that it is rather difficult to argue about “the very best set” of semantic categories it is more important that we could use the same concepts of flat category representations for a different corpus without major changes.

| Module | RTC Corpus | | BMC Corpus | |
|---|---|---|---|---|
| | correct assignments | | | |
| | training | testing | training | testing |
| BAS-SYN-DIS | 99% | 93% | 97% | 89% |
| BAS-SEM-DIS | 96% | 84% | 96% | 86% |
| ABS-SYN-CAT | 91% | 85% | 91% | 84% |
| ABS-SEM-CAT | 81% | 77% | 87% | 83% |
| PHRASE-START | 93% | 89% | 95% | 90% |

Table 5: Performance on the RTC and BMC Corpus

The two left columns of table 5 show the results of training and testing our five network architectures with the categories from the BMC corpus. In general, the results are comparable for the two corpora, although the individual network performance is slightly different. The two corpora are related in that they both contain spontaneous rather than written language and they contain interactions rather than monologs. On the other hand, interactions at a railway counter differ substantially in vocabulary, syntactic and semantic constructions. Furthermore, the BMC corpus contains less repairs, but the syntactic and semantic constructions are more complex due to more polite interactions in business contacts.

We consider our results as very promising since we have used spontaneous “noisy” language from dialogs rather than written language from texts. Furthermore, we use four flat learned representations, including two semantic ones, while previous learning approaches primarily focused on syntactic processing of texts. Based on these results we believe that learning a flat representation has the potential to increase portability in systems for spontaneous language analysis.

# 5 An example for running SCREEN

In this section, we demonstrate the incremental processing of the SCREEN system and focus in particular on the categorization part. The results of the incremental parse of a sentence are illustrated by snapshots of the running system at different times. In general, the system can process a large number of parallel sentence hypotheses produced by the speech recognizer. However, here we concentrate only on the single optimal sentence hypothesis for a short sentence. As an example we choose the sentence:

```
Yes I need eh a a sleeping-car <PAUSE>
   from <PAUSE> Regensburg to Hamburg.
```

The system snapshots show the state of the categorization part of SCREEN after a particular word hypothesis. The first snapshot in figure 2 shows the state after the fourth word hypothesis. However, in general, there could be multiple parallel sentence hypotheses of different length which we could view by scrolling up/down and right/left in the categorization window of sentence hypotheses. This categorization window of SCREEN shows the activation of the basic syntactic, the abstract syntactic, the basic semantic, and the abstract semantic category of a word hypothesis using a scaled square each as well as the phrase start using a scaled rectangle. The abbreviations below the squares correspond to the category abbreviations in tables 1 to 4. For instance, in the first snapshot we can see that the most favorite interpretation for the word hypothesis "I" is that it starts a new phrase, that the favorite basic syntactic category is a pronoun (U), the abstract syntactic category is a noun group (NG), that the basic semantic category is animate (ANIM) and the abstract semantic category is AGENT.

The size of the squares corresponds to the activation of the network category with the highest activation, but one can also inspect the activation of competitive categories by clicking on the activation values. For the favorite pronoun category (U) for the word "I" this second window is shown below the categorization window. Here we have an option to see the competitors of the favorite network activation at a more detailed level and we see that the second highest activation after the pronoun category was the noun category (N), which is quite reasonable at the start of the sentence hypothesis. The artificial neural network for BAS-SYN-DIS with the activations at this time is shown in figure 3. Each unit of the input layer represents a basic syntactic category as retrieved from the lexicon. This activation as well as the context is propagated through the hidden layer and finally to the output layer. The output layer possesses the same number of units as the input but the output units represent the

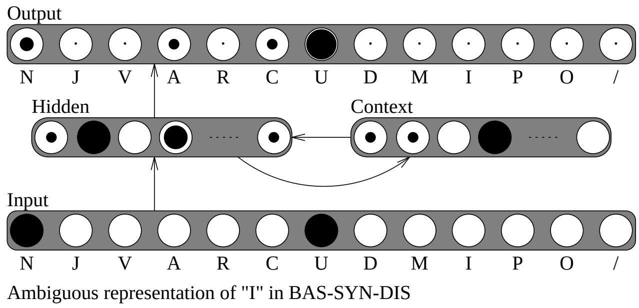


Figure 3: Recurrent network for basic syntactic categorization

disambiguated categories. The label of the output unit with the highest activation is taken as disambiguation result.

Furthermore, at the end of the sentence part in figure 2 we can see the interjection "eh" but without any favorite activation value. At this point only the word hypothesis has reached SCREEN, while the basic categories for the previous word "need" have already been computed, as well as the abstract categories for the second previous word "I". This illustrates the parallel interleaved and incremental processing in SCREEN. Interjections like "eh" belong to the hesitation phenomena which will be deleted for providing a better overall interpretation of a sentence. Therefore, in the second snapshot of figure 2 we see that the initially occurring interjection has been detected and eliminated from the original sentence hypothesis.

Furthermore, a possible repetition of constituents like word repairs can be dealt with. As an example, we illustrate the repetition of the determiner "a" in "yes I need a a". In general, the correction part in SCREEN can eliminate repairs based on the combined graded equality of the words themselves as well as their basic syntactic and basic semantic categories [11]. In this case, the words and the categories of the two words "a" correspond so that this repetition can be eliminated later. As another example for the more detailed inspection of categories in figure 2, we show the competitive activations for the hypothesis for the noun group (NG) for the first "a". The second snapshot shows that NG is clearly preferred compared to the other abstract syntactic categories of "a". Figure 4 gives a more detailed view of the ABS-SYN-CAT module for this situation. Each unit of the input layer represents a basic syntactic category. This input and the context are propagated to the hidden layer. The units of the hidden layer are copied to the context and propagated to the output layer. In the output layer each unit represents an abstract syntactic category and the one with the highest activation is selected.

The third snapshot of figure 2 shows that the word repetition with "a" has been eliminated. SCREEN also started to work on the word hypothesis for "sleeping-car" which has been clearly identified as a noun (N) as the basic syntac-

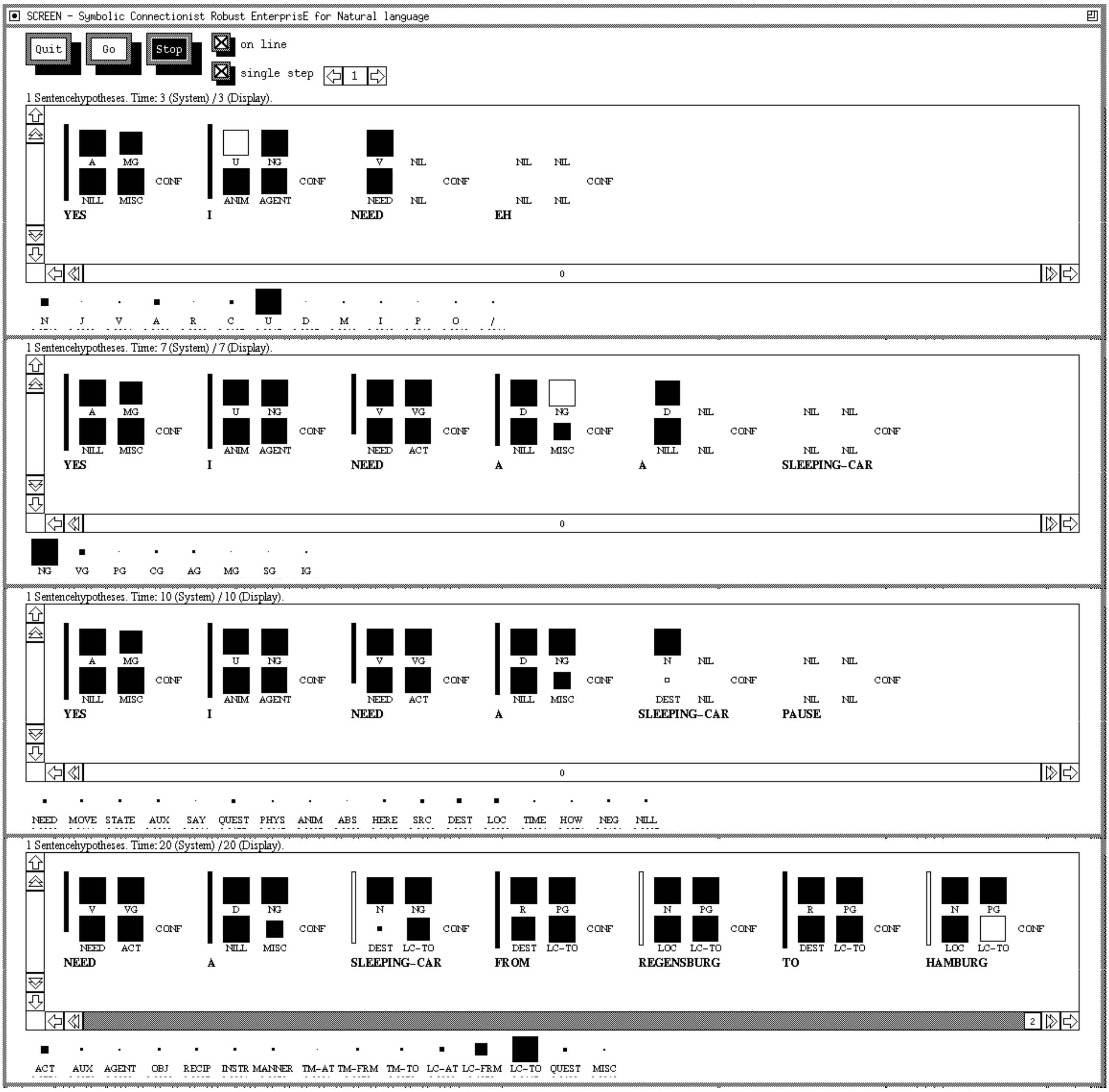


Figure 2: System snapshots of the category part. The abbreviations of the categories are explained in table 1 to 4.

tic category but which also has a very weak preference for a destination (DEST). Our inspection reveals that also all other competitive semantic categories have very low values close to 0. So the network has only very low preferences in this case, since it has not seen enough training instances of PHYSICAL objects (sleeping car) after NEED events. Therefore, the underlying network for basic semantic categorization could not learn a clear preference for "sleeping-car" as being PHYSICAL.

Our final system snapshot of figure 2 shows the end of the sentence. The previously appearing pause has been eliminated. Also the necessity of the phrase start can be clearly demonstrated here. Since we parse incrementally we associate each word with an abstract category. We take the first abstract syntactic category of a phrase group as the final abstract syntactic interpretation indicated by the filled rectangle phrase start marker since the first word like a preposition is a very good detector for an abstract syntactic category (based on the results in [15]). On the other hand, we take the last abstract semantic category of a phrase group since the last word like a head noun is a very good detector for an abstract semantic category (based on the re-

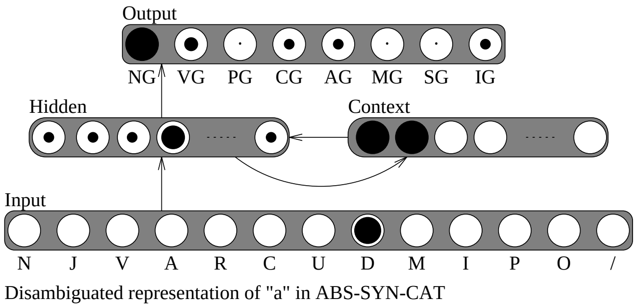


Figure 4: Recurrent network for abstract syntactic categorization

sults in [13, 16]). Phrase starts are also needed for distinguishing the same subsequent phrases. Already at the word "from" the abstract syntactic hypothesis favors the prepositional group PG. This favorite preference is kept for the subsequent words "Regensburg to Hamburg". If there would be no phrase start marker before "to Hamburg" this phrase could not be detected as a separate prepositional group.

Furthermore, we can see that "Regensburg" has been detected as a noun in a prepositional group, and is a location from which to start. In contrary, "Hamburg" in a different context, has been detected as a noun in a prepositional group, and is a location to which to travel. For the abstract semantic category (LC-TO) we also show the other competitors of the network (shown in figure 2) and we can see that - besides the favorite LC-TO - the second best preference would be LC-FRM, a second choice which is quite reasonable and understandable for city names. Only by the preceding context - either "from" or "to" - it is possible for the underlying network to make a decision whether the city name is a source or destination. Therefore, this example also illustrates the necessity of using recurrent networks with a memory for learning the preceding context.

# 6 Discussion and Conclusions

We have described the underlying concepts of flat representation, learning, and fault tolerance for processing spontaneous language in the novel system SCREEN. The system which is probably closest to our work is the PARSEC system [3]. PARSEC is a modular connectionist system which has been designed for the domain of conference registration. Input to the system is a sentence and the output is a connectionist case role analysis for the sentence. PARSEC is used in the speech translation system JANUS (e.g. [10]) as a backup-component if a symbolic parser does not provide a desired analysis. Rather than having multiple connectionist and symbolic parsers we integrate symbolic and connectionist properties in a single system; for instance connectionist representations support our category learning while symbolic representations are most useful for simple correction comparisons and for inter-module communication.

Currently, about four person years have been invested into the development of the novel concepts of SCREEN, the design of the overall architecture, the labeling of the two corpora, and the implementation of the networks, communication and interface. The category part, the core part of the system, has been fully implemented, trained and tested for the two different described domains. Furthermore, the modules for the correction part have been implemented and we also integrated a comprehensive implemented part for speech sequence construction. Sentences like the example sentence from figure 2 can be processed close to real time on a Sparc 2 machine. However, space limitations restrict us from describing all parts of SCREEN in detail, so that we focused in particular on the categorization part as the core part for demonstrating the underlying concepts for multiple corpora.

We believe that this work suggests a number of new concepts. First, neural network learning is an essential property in our system. Learning is not only useful for reducing knowledge engineering but learning also introduces a data-driven fault tolerance by using inductive learning on real-world data. Furthermore, we have a possibility to transport concepts to new domains without manually engineering syntactic or semantic rule or knowledge bases. Second, previous early work in artificial neural networks for language has often focused on small networks for *artificially generated training examples* (e.g., [2, 4, 5, 6]). Our work is using neural network based architectures on *real-world* spontaneous language. Third, we use preferences for different syntactic and semantic hypotheses. Besides a favorite interpretation there are always additional lower-rated interpretations. Fourth, the use of flat syntactic and semantic analysis vectors does not generate deep tree representations and has advantages for incremental processing of noisy spontaneous language. Using flat analysis SCREEN will *always* produce an interpretation and will not break on unusual syntactic or semantic constructions. We claim that the general properties of learning, fault tolerance, and flat representation have the potential to increase the portability of systems for spontaneous language analysis and we demonstrated this potential in our system SCREEN with real-world spoken language for two different domains.

**Acknowledgments**

This research was funded by the German Federal Ministry for Research and Technology (BMBF) under Grant #01IV101A0 and by the German Research Community (DFG) under contract DFG Ha 1026/6-2. We would like to thank S. Haack, M. Löchel, M. Meurer, U. Sauerland, and M. Schrattenholzer for their work on SCREEN.

# References


[1] *Neurocomputing – Foundations of Research*, eds., James A. Anderson and Edward Rosenfeld, The MIT Press, Cambridge, Massachusetts, 1988.

[2] Jeffrey L. Elman, `Finding structure in time', *Cognitive Science*, **14**(2), 179–211, (1990).

[3] Ajay N. Jain, `Generalization performance in PARSEC - a structured connectionist parsing architecture', in *Advances in Neural Information Processing Systems 4*, eds., John E. Moody, Steve J. Hanson, and Richard R. Lippmann, 209–216, Morgan Kaufmann, San Mateo, CA, (1992).

[4] James L. McClelland and Alan H. Kawamoto, `Mechanisms of sentence processing: Assigning roles to constituents of sentences', in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, eds., James L. McClelland, David E. Rumelhart, and The PDP research group, volume 2., Psychological and Biological Models, chapter 19, 272–331, MIT Press, Cambridge, MA, (1986).

[5] Clayton McMillan, Michael C. Mozer, and Paul Smolensky, `Rule induction through integrated symbolic and subsymbolic processing', in *Advances in Neural Information Processing Systems 4*, eds., John E. Moody, Stephen J. Hanson, and Richard P. Lippmann, 969–976, Morgan Kaufmann, San Mateo, CA, (1992).

[6] Risto Miikkulainen, *Subsymbolic Natural Language Processing. An integrated model of scripts, lexicon and memory*, MIT Press, Bradfor Book, Cambridge, MA, 1993. ISBN 0-262-13290-7.

[7] Michael C. Mozer, `Neural net architecture for temporal sequence processing', in *Predicting the future and understanding the past*, eds., A. Weigend and N. Gershenfeld, Addison-Wesley Publishing, Redwood City, CA, (February 1993).

[8] *Connectionism in Perspective*, eds., R. Pfeifer, Z. Schreter, Françoise Fogelman, and L. Steels, North-Holland, Amsterdam, Netherlands, 1989.

[9] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams, `Learning internal representations by error propagation', in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, eds., David E. Rumelhart, James L. McClelland, and The PDP research group, volume 1., Foundations, chapter 8, 318–362, MIT Press, Bradford Books, Cambridge, MA, (1986). also published in: [1, Chapter 41, pp. 675–695].

[10] Alexander Waibel, Ajay N. Jain, A. McNair, J. Tebelskis, L. Osterholtz, H. Saito, O. Schmidbauer, T. Sloboda, and M. Woszczyna, `JANUS: Speech-to-speech translation using connectionist and non-connectionist techniques', in *Advances in Neural Information Processing Systems 4*, eds., John E. Moody, Steve J. Hanson, and Richard R. Lippmann, 183–190, Morgan Kaufmann, San Mateo, CA, (1992).

[11] Volker Weber and Stefan Wermter, `Artificial neural networks for repairing language', in *Proceedings of the 8 $^{th}$ International Conference on Neural Networks and their Applications (NEURAP'95)*. (Marseilles, FRA), (December 1995).

[12] Volker Weber and Stefan Wermter, `Towards learning semantics of spontaneous dialog utterances in a hybrid framework', in *Hybrid Problems, Hybrid Solutions — Proceedings of the 10$^{th}$ Biennial Conference on AI and Cognitive Science (AISB-95)*, ed., John Hallam, pp. 229–238. (Sheffield, UK), (1995).

[13] Stefan Wermter, `Learning natural language filtering under noisy conditions', in *Proceedings of the Tenth IEEE Conference on Artificial Intelligence for Applications*, pp. 941–944, (1994).

[14] Stefan Wermter, *Hybrid Connectionist Natural Language Processing*, Chapman and Hall, London, UK, 1995.

[15] Stefan Wermter and Matthias Löchel, `Connectionist learning of flat syntactic analysis for speech/language systems', in *Proceedings of the International Conference on Artificial Neural Networks (ICANN'94)*, eds., Maria Marinaro and Pietro G. Morasso, volume 2, pp. 941–944, London, (May 1994). (Sorrento, Italy), Springer-Verlag.

[16] Stefan Wermter and Ulf Peters, `Learning incremental case assignment based on modular connectionist knowledge sources', in *Proceedings of the World Congress on Neural Networks*, eds., Paul Werbos, Harold Szu, and Bernard Widrow, volume 4, pp. 538–532, Hillsdale, NJ, (June 1994). INNS, (San Diego, CA), Lawrence Erlbaum Associates.

[17] Stefan Wermter and Volker Weber, `Learning fault-tolerant speech parsing with SCREEN', in *Proceedings of the 12 $^{th}$ National Conference on Artificial Intelligence (AAAI-94)*, volume 1, pp. 670–675, Menlo Park, (July/August 1994). Seattle, Washington, AAAI Press/The MIT Press.