FROM J. FISCHER ET AL., *SPECOM'99, INT. WORKSHOP SPEECH AND COMPUTER,* MOSCOW, 1999, PP. 75-78.

COMBINING STATISTICS WITH SEMANTIC NETWORKS IN A REAL-TIME DIALOGUE SYSTEM

J. Fischer, E. Nöth, H. Niemann

Universität Erlangen-Nürnberg, Lehrstuhl für Mustererkennung (Informatik 5)

Martensstraße 3, D-91058 Erlangen, Germany

email: fischerj@informatik.uni-erlangen.de Tel.: +49/9131/8527824 Fax: +49/9131/303811

Abstract: We use an innovative approach to speech understanding which is based on a fine-grained knowledge representation automatically compiled from a semantic network, and on an iterative control strategy. This approach, besides allowing an efficient exploitation of parallelism, enables the system with any-time capability since after each iteration a (suboptimal) solution is available. We make use of statistical methods (e.g. neural networks, n-grams, classification trees, HMMs, and a frequency based method) to improve the efficiency of the system and to learn linguistic restrictions in order to simplify the adaption of the system to new application domains. We show the feasibility of our approach by experimental results.

1 INTRODUCTION

In order to make use of automatic speech understanding systems in real world applications, those systems have to be featured with real-time and any-time capabilities. Furthermore, they should be easily adaptable to new application domains, and should be able to integrate speech with other sources of information, for instance gestures, which would increase the accuracy and naturalness of human computer interaction. The combination of statistical methods with knowledge based methods combined with parallel and iterative processing in a formalism which allows both, speech and image understanding, seems to be a promising means to achieve the capabilities mentioned above.

Whereas the use of statistics on the level of speech recognition is state-of-the art, speech understanding is usually based on parsing algorithms and context-free grammars. In the past few years, statistical methods gained more and more importance also on the level of understanding (see for instance [9, 10]). A variety of parallel algorithms for problems from data-driven processing have been developed. In contrast, parallel symbolic processing is much less investigated, although some major problems of the field, like e.g. parallel knowledge representation [3], are discussed in the literature. In our approach, we combine knowledge based speech understanding using semantic networks with statistical methods to speed-up the search for the best interpretation. The semantic network provides an integrative knowledge representation formalism for speech and image understanding. By using statistical methods, a fast adaptability to new application domains is enabled, provided that a corpus of data is available. Furthermore, a control algorithm based on parallel iterative optimization is used, providing the desired any-time and real-time behaviour.

2 THE DIALOGUE SYSTEM

As a framework for our approach we use the dialogue system EVAR [8] which answers queries about the German train timetable. The linguistic knowledge representation of EVAR is arranged in 5 *levels of abstraction*: The *Word hypotheses* level represents the interface between speech recognition and speech understanding; on *Syntax* level syntactic constituents are represented; the *Semantic* level is used to model verb and noun frames with their deep cases for task independent interpretation; on *Pragmatic* level, semantic information is interpreted in a task specific context; the *Dialog* level models possible sequences of dialogue acts. The knowledge itself is represented using the semantic network formalism of ERNEST (ERlanger NEtzwerk SysTem). Knowledge about general terms, events, etc. is represented in *concepts* C (e.g. SY_NOUN represents knowledge about nouns), actual realizations of a concept are represented by *instances* $I(C)$ (e.g. the actual word hypothesis for the noun *"train"* is represented by an instance $I(\text{SY\_NOUN})$). Relations between the concepts (nodes) are established by *part*, *concrete*, and *specialization links*. The main components of a concept C itself are, besides its *parts* P and *concretes* K, a set of *attributes* A and *structural relations* S. Each of them references a function F which computes the value of the corresponding attribute and a measure of the degree of fulfillment of the relation. Since there may be different possibilities for the actual realizations of a concept and in order to allow a compact knowledge representation, *modalities* H_l are introduced with the implication that each individual modality $H_l^{(k)}$ may define the concept C_k. A noun phrase, for example, can be defined by modality 1: *proper noun* (obligatory part), e.g. *"Berlin"*, or modality 2: *noun* (obligatory part), *article* and *adjective* (optional parts), e.g. *"the next train"*. For the computation of an instance $I(C)$, instances for at least all obligatory parts and concretes occurring in one modality of C, and values for all attributes and relations of C have to be computed. Because of multiple occurrences of the same words (or word categories) in the set of word hypotheses and ambiguities in the knowledge base (arising from the various modalities), competing instances may be computed. Thus, a *confidence measure* G is available, which computes the degree of confidence of an $I(C)$ and its expected contribution to the success of the analysis. The ultimate goal of the analysis is represented by a *goal concept* C_g, which represents the demanded symbolic description of the word hypotheses. Thus, an interpretation of the word hypotheses is given by

an instance of the goal concept $I(C_g)$. Task of the control algorithm is to search for an *optimal instance* $I^*(C_{g_i})$ with highest confidence value, which corresponds to the best interpretation.

Our control algorithm [4] treats the search for an optimal interpretation as a *combinatorial optimization* problem and solves it by means of *iterative* optimization methods. For this purpose, a state of analysis (or interpretation) vector $\boldsymbol{r}$ is introduced, which makes the assignment of exactly one modality to each ambiguous concept C_k, and one word hypothesis O_j to the primitive attribute A_i (see below) of each concept on word hypotheses level:

$$\boldsymbol{r} = [(A_i, O_j^{(i)}); (C_k, H_l^{(k)}) \mid i = 1, \ldots, m \ ; \ k = 1, \ldots, n] \ . \tag{1}$$

Each specific value allocation of this vector reflects exactly one out of all possible interpretations. All possible value allocations of this vector build the *search space* of the analysis. Once a specific value allocation (i.e. a current state of analysis $\boldsymbol{r}_c$) has been chosen, the corresponding interpretation can clearly be computed in a bottom-up way. In order to make this computation more efficient and to allow an efficient exploitation of parallelism, the concept centered and well structured knowledge base is automatically and off-line compiled into a fine-grained task graph (the so-called *attribute network*, see Fig. 1). It explicitly represents the dependencies of all attributes, relations, and confidence measures of all instances of concepts to be computed for an interpretation.

The attributes, relations, and confidence measures are represented by the nodes, the dependencies between them by the directed links of the task graph. Nodes without predecessors (*primitive attributes*) build the interface to the word hypotheses and nodes without successors are the confidence measure of the goal concept. Now, in each iteration a current state of analysis is chosen and mapped onto the attribute network. The corresponding interpretation and its confidence value are then computed by a bottom-up processing of all nodes of the attribute network. Iteration steps are performed until the 'best' interpretation is found or the specified processing time for analysis has been used up. *Parallelism* can be exploited on two levels: on *knowledge level* by a parallel computation of attribute network nodes on the same layer, and on *control level* by the computation of competing instances on several workstations (for example, by means of PVM). Notice that, when interpretation is stopped due to time limitations, a (suboptimal) interpretation is always available if at least one iteration – which at the moment needs less than 0.2 seconds of processing time – was performed (any-time capability).

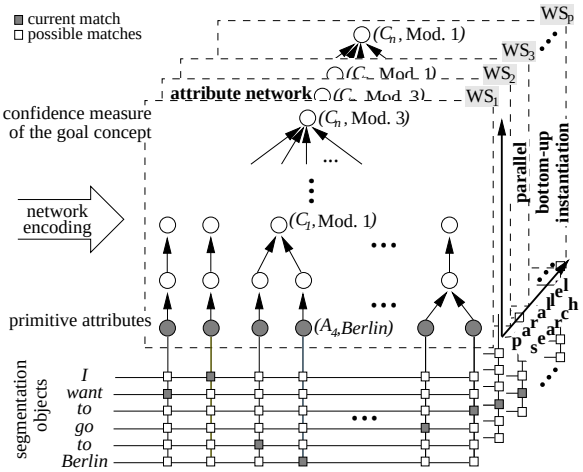

Figure 1: Scheme of the parallel iterative control algorithm.

Figure 1 shows an example for the analysis of the word chain "*I want to go to Berlin*". The current state of analysis for which an instance is computed bottom-up on the first workstation (WS_1) is

$$\boldsymbol{r}_{cWS_1} = [(A_1, \textit{want}), \ldots, (A_4, \textit{Berlin}), \ldots, (A_m, \textit{to}); (C_1, \text{Mod.1}), \ldots, (C_n, \text{Mod.3})],$$

with C_n being the goal concept C_g. The current state of analysis on WS_2 differs from that on WS_1 at least by the current word hypothesis assigned to A_m ("*go*"); the current state of analysis on WS_p differs from that on WS_1 and WS_2 at least by the modality assigned to C_n, indicating that competing instances are computed for different states of analysis on the various workstations.

3 COMBINING STATISTICS WITH SEMANTIC NETWORKS

Initialization If an initial state of analysis $\boldsymbol{r}_0$ is chosen at random, it often occurs that the algorithm starts searching quite far from the optimal solution and thus needs a lot of iterations (and hence a great deal of processing time) until the optimum is found. Thus, in order to find the best interpretation in an efficient manner, the careful choice of an initial state of analysis is indispensable. For example, if a noun phrase for the word hypotheses "*the next train*" is to be instantiated, and modality 1 is chosen to compute the corresponding $I(\text{SY\_NP})$ (cf. Section 2), the confidence value of this instance will be quite low or "invalid".

We make a prediction of the optimal state of analysis by means of the given word chain $\boldsymbol{w}$ (which can be the best word chain computed from the word graph) using time delay neural networks (TDNNs) [1], semantic classification trees (SCTs) [7], and n-gram language models. Since the number of word hypotheses for each primitive Attribute A_i varies for each $\boldsymbol{w}$ to be analyzed (according to the number of concurring hypotheses for each A_i), we employ these methods for the initialization concerning the assignment of a modality for each ambiguous concept C_k. An initialization concerning the assignment of word hypotheses to primitive attributes is realized in connection with the learning of linguistic restrictions as explained below.

We use a method for categorization which allows to represent several features of a single word in one category. Therefore, we define the category itself as consisting of a *basic part* and a *special part*. The basic part contains syntactic information about the word, the special part semantic information. Consider, for example, the word "*Intercity*". It's basic category is NOUN, it's special category is TRAIN. The resulting category system is not disjoint and used for the TDNNs. Each category is represented as a binary vector $\boldsymbol{i}$, which serves as *input* for the TDNN. If a word w is

part of two or more categories, the appropriate input vector results from a combination of the binary representation of these categories. The structure of the input vectors is chosen in such a way that the combination results in a unique representation of the categories. The *output* $\boldsymbol{o}$ of the TDNN is also a binary vector which is mapped to the initial state of analysis vector $\boldsymbol{r}_0$.

As the number of nodes to be initialized is very high and it is not feasible to train a classifier based on SCTs or n-grams for each node, we reduce the task for SCTs and n-grams to some very important nodes in the attribute network, in order to '"support'" the results of the TDNN. Therefore we concentrate on the initialization of the verb frame which gives us ten different classes and use a system with 71 disjoint categories which also comprise syntactic and semantic information. As output we get the verb frame we have to choose for our initial state of analysis vector. For the classification of word chains with n-gram language models we proceed as follows: For each class to be distinguished we train a separate n-gram on those sentences from the training set corresponding to the according class. In the case of classifying verb frames we train ten different language models. For a new sentence to be analyzed we compute the different probability scores of the models and decide for that with highest probability. For more details see [5].

Learning Domain Dependent Knowledge In order to allow an efficient linguistic analysis, the enormous search space has to be reduced. This is done by making use of specific linguistic knowledge about syntax, semantics and pragmatics in the way of *linguistic restrictions*.

These restrict the range of values for the attributes A^j (e.g. case, gender, semantic class, etc.) of concepts from all possible values $A^j_q,\ q = 1, \ldots, N^j$ to a few ones. For example, the attribute gender of a word can be restricted from *masculine, feminine, neuter* to only *feminine*. These restrictions are usually entered manually into the respective attribute-slots of the concepts in the knowledge base. The attribute network of our approach allows us to propagate top-down (from dialogue level to the primitive nodes on the word hypotheses level) those restrictions concerning the properties of the word hypotheses once before the analysis starts. This enables the rejection of concurring word hypotheses during analysis which could potentially be assigned to a primitive node but which violate at least one of the restrictions at this node. The reduction of search space is thus achieved by a drastic reduction of the number of competing word hypotheses at each primitive node. In our approach we learn these linguistic restrictions at each primitive node within the attribute network by adding a learning component to the control of the analysis. This learning component is based on the counted frequencies of attribute values which were observed at the primitive nodes of the attribute network and those which succeeded (i.e., attribute values of that word hypothesis which finally led to the best interpretation), using a set of training utterances. To make use of the learned restrictions we defined a *rating function* $\bar{J}_i(l)$ which during analysis computes at each primitive attribute node i a rating for each word hypothesis $w_i(l), l = 1, \ldots, L$ (number of concurring word hypotheses at node i). This rating is based on the learned linguistic restrictions and on contextual information provided by a global classifier, which is realized by means of generalized hidden markov models (GHMMs, cf. [6]). This rating represents a measure of the compatibility of a word hypothesis with a primitive node. It can be further improved online after each new analyzed utterance. The ratings are used as follows to speed up the analysis: They enable an *initialization* of the state of analysis vector by assigning an "optimal" word hypothesis to each primitive node. Furthermore, they allow a *weighted* change of word hypotheses during the optimization, when a new current state of analysis is chosen before a new iteration step. Figure 2 illustrates the principle of the learned ratings. In this figure $i = 1, \ldots, m$ and node m has the role of a preposition (on pragmatic level, this primitive node belongs to a *place of arrival*). Thus, all words of the word lattice which are prepositions ("to", "from", and "to") are concurring at node m.

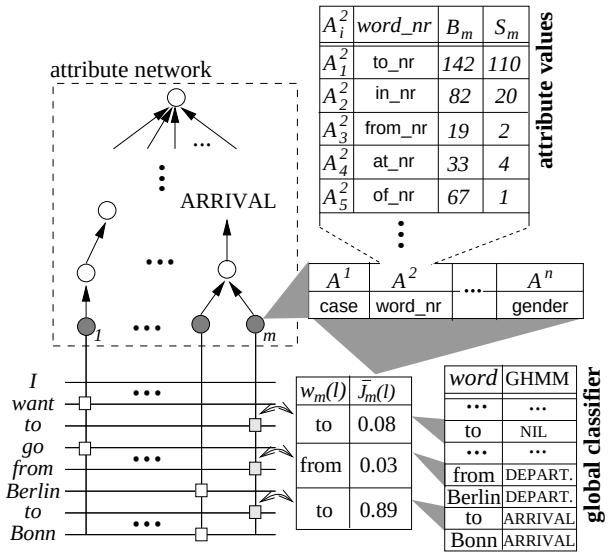

Figure 2: Principle of the learned rating function.

The learned linguistic restrictions for the attribute values (case, word_nr, ..., gender) at node m are illustrated in the upper right, being B_m the number of observed attribute values and S_m the number of succeeded attribute values counted at node m during the analysis of the training utterances. Together with the results of the global classifier (lower right) which delivers coarse contextual information by means of the best word chain, a rating $\bar{J}_m(l)$ for each preposition is computed: "to" = 0.08, "from" = 0.03, "to" = 0.89. As expected, the second "to" has the highest rating, since it is in fact the preposition of the place of arrival "to Bonn". Without contextual information, the ratings would be 0.89, 0.33, and 0.89, respectively. For more details see [2].

4 EXPERIMENTAL RESULTS

For our experiments we use the transliteration of 6 712 *spontaneous* utterances selected from a corpus (simulating a 100% word accuracy) collected via the public telephone network. Training is done with 5 767, tests with 945 of

these utterances. The labeling of the test data for the training of the TDNN was done automatically by the system itself, using a heuristic initialization which consists of a small set of rules working on the incoming word chain and performing 25 iterations. For the training of the SCT and n-gram, whose task is only to adjust the verb frame, the labeling with verb frames was done semi-automatically: a small set of rules classified the word chain in a first step and in a second step we checked and corrected the results. The trained TDNN consists of 42 input nodes (we enter two words at a time) and 542 output nodes. There are five hidden layers, each consisting of 70 nodes, and a context of 16 words can be considered. The attribute network we compiled out of the knowledge base for the evaluation consisted of about 10 000 attribute nodes. Evaluation was done regarding the amount of correct pragmatic units (e.g. *place of arrival, time, date,* etc.) found.

The experiments have shown that the initialization with the TDNN even outperformed the heuristic initialization (which led to an error reduction of 68%). Since the computing time for the statistical initialization ($\approx$ 0.4 seconds) is about the same as for performing two iterations (one iteration needs $\approx$ 0.2 seconds on a 9000/735 HP Workstation), and since it corresponds to the state of analysis after 25 iterations without initialization, we drastically accelerate the analysis. At the moment, the system's real-time factor for the interpretation of the initial user's utterance is 0.7, performing 5 iterations and simulating parallel processing on 5 processors. We achieved the best result by employing the TDNN combined with a verb frame initialization computed by the SCT: 83.1% after 5 iterations (with the heuristic initialization we achieved 79%). This can be explained by the fact that verb frames in German often are defined by keywords which can have long distance dependencies between them. These can be modeled with the SCT but not with the n-gram. Furthermore, the verb frame initialization by the SCT can intercept errors made by the TDNN, and thus lead to an overall better result.

Concerning the learning of linguistic restrictions, we evaluated the performance of four different system variants. *System 1* does not contain any restrictions and initializes the primitive nodes with randomly chosen word hypotheses; *System 2* does not contain any restrictions but uses a *heuristic* initialization for the assignment of word hypotheses to primitive nodes; *System 3* contains the manually entered restrictions and uses the heuristic initialization; *System 4* is based exclusively on the confidence values introduced in section 3, both for initialization and weighted changes of word hypotheses. This system variant was trained by removing the existent, manually modelled restrictions out of system 3 and learning them from scratch by means of the set of classified utterances. For the experiments we used the same spontaneous utterances as mentioned above, training is done with 6385 utterances and for testing we use the remaining 327 ones. The percentage of the correctly analyzed pragmatic unit *place of arrival* is 48.8%, 76.6%, 94.6%, and 95.6%, for the four systems respectively, and 25 iterations. System 4, which utilizes the new learned confidence values is superior to all the other systems. This proofs that our approach is able to learn the linguistic restrictions and that the confidence values are doing very well for the initialization and the weighted change of word hypotheses.

5 FUTURE WORK

Our future work will concentrate on the integration of the methods presented here in one system (at the moment, we have two versions of the system on which we made the experiments). As regards the learning of linguistic restrictions, there are still modifications of the procedural knowledge necessary in order to take full advantage of the approach's potential. Furthermore, the parallel processing will be implemented by means of the *parallel virtual machine* (PVM).

REFERENCES

[1] Y. Bengio. *Neural Networks for Speech and Sequence Recognition*. International Thomson Computer Press, London, 1995.

[2] F. Deinzer, J. Fischer, U. Ahlrichs, and E. Nöth. Learning of Domain Dependent Knowledge in Semantic Networks. In *EUROSPEECH '99. Proc. of the 6th European Conf. on Speech Communication and Technology*, Budapest, 1999. *To appear.*

[3] M. Evett, J. Hendler, and L. Spector. Parallel Knowledge Representation on the Connection Machine. *Journal of Parallel and Distributed Computing*, 22(2):168–184, 1994.

[4] V. Fischer and H. Niemann. A Parallel Any–time Control Algorithm for Image Understanding. In *Proceedings of the* 13^{th} *International Conference on Pattern Recognition (ICPR)*, Wien, October 1996. IEEE Computer Society Press.

[5] Fischer, J. and Haas, J. and Nöth, E. and Niemann, H. and Deinzer, F. Empowering Knowledge Based Speech Understanding through Statistics. In *ICSLP*, volume 5, pages 2231–2235, Sydney, Australia, dez 1998.

[6] J. Haas, J. Hornegger, E. Nöth, and H. Niemann. A Probabilistic Approach for the Semantic Analysis. In *Proc. of the AIII Workshop on Artificial Intelligence in Industry*, pages 422–430, 1998.

[7] R. Kuhn. *Keyword Classification Trees for Speech Understanding Systems*. PhD thesis, School of Computer Science, McGill University, Montreal, 1993.

[8] M. Mast, F. Kummert, U. Ehrlich, G. Fink, T. Kuhn, H. Niemann, and G. Sagerer. A Speech Understanding and Dialog System with a Homogeneous Linguistic Knowledge Base. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(2):179–194, 1994.

[9] R. Pieraccini and E. Levin. A Learning Approach to Natural Language Understanding. In *NATO-ASI, New Advances & Trends in Speech Recognition and Coding*, volume 1, pages 261–279, Bubion (Granada), Spain, 1993.

[10] H. Stahl, J. Müller, and M. Lang. An Efficient Top–down Parsing Algorithm for Understanding Speech by using Stochastic Syntactic and Semantic Models. In *Proc. Int. Conference on Acoustics, Speech, and Signal Processing*, pages 397–400, Atlanta, 1996.