LEARNING COMPLEX OUTPUT REPRESENTATIONS IN CONNECTIONIST PARSING OF SPOKEN LANGUAGE *

Finn Dag Buø[1] *Thomas S. Polzin*[2] *Alex Waibel*[1,2]

[1]University of Karlsruhe, Karlsruhe, Germany
[2]Carnegie Mellon University, Pittsburgh, PA, USA

ABSTRACT

Due to robustness, learnability and ease of integration of different information sources, connectionist parsing systems have proven to be applicable for parsing spoken language, However, most proposed connectionist parsers do not compute and represent complex structures. These parsers assign only a very limited structure to a given input string. For spoken language translation and data base access, more detailed syntactic and semantic representation is needed. In the present paper, we show that arbitrary linguistic features and arbitrary complex tree structures can indeed also be learned by a connectionist parsing system.

1. INTRODUCTION

The connectionist parser of spoken language, PARSEC, [4, 5, 7, 8] has been proven to be successful for particular problems frequently found when processing speech. First, PARSEC is robust towards ungrammaticality, restarts, and other spontaneous speech effects. Second, PARSEC is trainable, and thus eliminates the effort of writing a grammar. Third, within PARSEC it is easy to integrate different information sources. For instance, information about the pitch contour is used to distinguish between statements and yes-no questions.

However, in comparison with symbolic parsing systems e.g. [1, 2, 3] that rely on hand encoded grammar rules, the output of many currently proposed connectionist parsers [5, 9, 10] is limited. In particular, morpho-syntactic and structural information is not part of the output structure of the current PARSEC system. Moreover, detailed structural relations are necessary in order to resolve anaphoric relations, i.e. finding the referent of pronouns. Consider for example the output from the old PARSEC system shown in figures 1 and 3.

Performing a high-quality speech translation based on this output is difficult (see figure 2), because no information about the *morpho-syntactic features* and *structural relationships* is available. For example, just looking in the lexicon to decide whether *artikel* (paper/s) is singular or plural is not sufficient, because the respective forms are identical. In the example above, the relevant information about number can only be decided in the actual context where *ihr* (your) and *wird* (to be) indicate a singular reading of *artikel*. Further, *werden* (base form of *wird*) signals passive in the first clause, and future in the second clause.

Morpho-syntactic features include number, tense, agreement, mood, gender etc. Some of these values may be taken from the lexicon, but most of them can only be decided in the actual context within the sentence.

Structural relationships include attachment of prepositional phrases, relative clauses, genitives, adjectives etc. Consider the output of the old PARSEC system shown in

```
([statement]
 ([sub-clause]
  ([misc]        falls)
  ([agent]       ihr artikel)
  ([action]      akzeptiert)
  ([iaux]        wird))
 ([clause]
  ([iaux]        werden)
  ([agent]       wir)
  ([recipient]   ihnen)
  ([adverb]      auch)
  ([patient]     spezielle formulare)
  ([mod-1]       fuer ihren artikel)
  ([action]      zusenden)))
```

Figure 1. Parse lacking features

If your paper is accepted,
we'll also send you special forms for your paper. (correct)
If your papers are accepted,
we'll also send you special forms for your paper.
If your paper will accept,
we'll also send you special forms for your paper.
If your paper is accepted,
we'll also send you special form for your paper.
If your paper is accepted,
we'll also send you special forms for your papers.

Figure 2. Some possible English translation errors of sentence parsed in figure 1, when ignoring feature information.

*The first author was supported by the Research Council of Norway. The second and third authors were supported by the Advanced Research Projects Agency and the National Science Foundation. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the offical policies, either expressed or implied, of the Advanced Research Projects Agency, the National Science Foundation, or the Research Council of Norway.

0-7803-1775-0/94 $3.00 © 1994 IEEE

figure 3.

```
([statement]
 ([sub-clause]
  ([agent]      his big brother+s friend)
  ([action]     loved)
  ([patient]    himself)))
```

Figure 3. Parse lacking structure

The important point to note here is that no internal analysis of the agent slot is performed by the system. In a machine translation framework the analysis above would not be sufficient to enable a regular mapping into the target language. A much more detailed analysis of the internal structure of e.g. the agent slot is needed.

Both problems above could be tackled by a symbolic rule based parsing systems. However, ungrammatical sentences (which are frequently found in spontanuous speech) represent a major problem. On the other hand, a connectionist system with its inherent robustness can compute this kind of input with less difficulty.

Moreover, the impoverished output of connectionist systems limited their application in larger systems (for instance the JANUS system), because they required heavy (symbolic) postprocessing. Further, in most cases, these postprocessors must be targeted strongly towards the application.

Although we think that hybrid architectures are quite useful, there is always the problem of linking two submodules and defining interface relations. In particluar, linking of connectionist modules to symbolic modules faces the additional problem of mapping continuous into discrete values. During this kind of mapping relevant information might be lost and additional sources of possible breakdowns might be introduced into the system.

In general, we think that the scope of one module should be as wide as possible. In the case of connectionist parsing systems this means that the connectionist system should assign a syntactic structure to an input sentence without relying on symbolic postprocessing.

2. LEARNING LINGUISTIC FEATURE VALUES

In a first approach we extended PARSEC is such a way that the system was able to learn to parse sentences into case frames annotated with any kind of linguistic features. The system was trained and tested with morpho-syntactic feature values, but semantic, pragmatic, and prosodic feature values would also have been possible.

Contrast the parse results of the old system in figure 1 with the output of the modified system in figure 4, where the feature pairs (pair of feature and feature value) are emphasized. They are given in the form (feature feature_value). Any number of feature pairs is allowed. PARSEC labels are shown in square brackets. They are included for completeness only.

From a theoretical point of view ([1], pp.35), these features are *atomic-valued* features, in contrast to *category-valued* features. Each atomic-valued feature takes a maximum of value 1 out of a class of m values. In a neural network paradigm, this means that we have a classifier.

To learn the task, normal connectionist backpropagation networks with one hidden layer were used. The old PARSEC system was used as input to these networks: Words were mapped to one vector of binary features per word. The sentences were segmented into phrases and clauses (cf. [5] for more details).

When finding the optimal architecture of the feature value networks, two slightly different approaches were tried:

First, each feature was learned by a separate connectionist network, i.e. each network is a separate classifier. This approach has the advantage that each network gets a relatively easy task to learn, and the disadvantage that it requires many networks and long training time.

Second, all features belonging to phrases were trained with the phrase label in one network together, i.e. each network is a collection of classifiers. This approach can exploit the correlation between features, and between features and the case label, but has the disadvantage that the task to learn is more complex.

Training and testing runs were made with German data from the Conference Registration Task, using 9 dialogs for training, and 3 other dialogs for testing. (To train and test on German is better than on English, because German has a richer morpho-syntax than English.) Training performance was 99.0 % and 97.7 % (averaged over the features), respectively for the two architectures. Testing performance was 87.3 % and 88.6 % (averaged over the features), respectively.

3. CONNECTIONIST SHIFT-REDUCE PARSING

In a second approach, we implemented (based on PARSEC) a connectionist system which was trainable to approximate the behavior of a symbolic bottom-up shift-reduce parser. The underlying architecture consists of two backpropagation networks. The first network is responsible for detecting phrase boundaries in the input string, the second for assigning labels to the respective phrases. In order to generate arbitrary complex tree structures we added a recursion step to the system: the list of labels assigned to the respective phrases in time step n will serve as the input string in time step $n+1$. Therefore, the labels (e.g. noun phrase (*NP*) or verb (*V*) have to be defined as lexical items in the lexicon as well. For the sentence *his big brother+s friend loved himself* the first recursion step will assign the preterminals *N ADJ N N V N* to the input string. This list of preterminals will be the input string in the next recursion step which computes *N-BAR ADJ-BAR N-BAR N-BAR V N-BAR* as its output. The recursion terminates if no more *shift* or *reduce* actions can be executed. Instead of the parse in figure 3, the system will now output the phrase structure tree in figure 5. We used a slighly modified X-bar grammar formalism [6] which was easier to learn by the network because of its uniformity in the underlying rule representation. Note that the phrase structure is not required to be in some normal-form (e.g. Chomsky normal-form).

The generation of training data now becomes an essential part of the system, because the training data finally determines the dynamic behaviour of the connectionist system.

```
([statement]
  ([sub]      ((form passive) (tense present)
               (mood ind) (agr sing_3))                                   ;features of verb clause 1
    ([misc]                                          falls)
    ([agent]  ((case nom) (agr sing_3) (gender masculine)) ihr artikel)   ;features of NP 1
    ([action]                                        akzeptiert)
    ([iaux]                                          wird))
  ([clause]   ((form active)
               (tense future) (mood ind) (agr plu_1))                     ;features of verb clause 2
    ([iaux]                                          werden)
    ([agent]  ((case nom) (agr plu_1))               wir)                 ;features of NP 2
    ([recipient]((case dat) (agr pol_2))             ihnen)               ;features of NP 3
    ([adverb]                                        auch)
    ([patient] ((case acc) (agr plu_3) (gender neuter))  spezielle formulare);features of NP 4
    ([mod-1]  ((case acc) (agr sing_3) (gender masculine)) für ihren artikel) ;features of PP 1
    ([action]                                        zusenden)
))
```

Figure 4. Case frames annotated with features

The process of generating training from an abstract representation for the example sentence is given in figure 6. It shows the respective input "sentences" for all levels in the respective syntactic tree (cf. figure 5), where to put phrase boundaries, and how to label the respective phrases.

From the resulting more detailed syntactic analysis of input sentences it is much easier to define a mapping into some target representation e.g. an interlingua for machine translation or an SQL for data base access. In addition, this more detailed structure can be used [11] to resolve anaphoric relations obtaining within the sentence. For example, the only possible referent for the reflexive pronoun *himself* in the sentence above is the noun phrase in the subject position: *his big brother+s friend*. However, in a sentence like *his big brother+s friend loved him* the pronoun *him* cannot have the entire subject noun phrase as its antecedent.

The system was trained with a corpus of over 100 sentences. It learned all sentences, and generalized well [11].

4. CONCLUSION

The two approaches described above show that connectionist systems are able to compute sophisticated and complex output structures. We extended the scope of connectionist parsing systems within an NLP system from some kind of *preprocessor* to a substantial part of the overall system architecture. As a consequence, the learning capabilities of connectionist systems can cover much more of the overall processing task (thereby eliminating the need for hand specified rule-based system components). In addition, because now the connectionist systems are able to output much more structured representations, it is easier to define interfaces with other (symbolic) system components.

Both approaches learned and generalized well from small corpora, which reduces the human effort to a minimum when making a running system for a specific application.

In future work, it would be interesting to combine structure and feature information in one system. We see at least two approaches to do this:
1. Combining the two methods described here.
2. Extending the first method to not only learn atomic-valued features, but also category-valued features. This would mean that the system would produce feature value structures, and hence be output compatible with unification based parsers, e.g. [2]. Ongoing work is is based on the second approach.

The first approach was trained and tested with only morpho-syntactic feature values, but is by no means limited to this kind of feature. One could also try learning semantic, pragmatic, and prosodic feature values. Ongoing work aims at testing the first method on the Time Scheduling Data Base. This data base is larger than the Conference Registration Task. Moreover, it is a true spontaneous task.

5. ACKNOWLEDGEMENTS

Part of the work of the second author was done within a graduate class in the Computational Linguistics Program at Carnegie Mellon. We would like to thank the instructor David A. Evans and the participants of the class for valuable discussions and comments.

REFERENCES

[1] G.Gazdar, E.Klein, G.K. Pullum, and I.A. Sag. *Generalized Phrase Structure Grammar*. Blackwell Publishing, Oxford, England and Harvard University Press, Cambridge, MA, USA, 1985.

[2] M.Tomita and J.Carbonell. *The universal parser architecture for knowledge-based machine translation*. Tec. report CMU-CMT-87-101, Center for Machine Translation, CMU, 1987.

[3] Hiroaki Saito. *Interactive Speech Understanding*. In *Proceedings of COLING*. Nantes, France, 1992.

[4] Ajay Jain and Alex Waibel. *Robust connectionist parser of spoken language*. In *Proceedings of the IEEE*

```
                       S
                       |
               ------------------
               |               |
              NP              VP
               |               |
               |             V-Bar
               |               |
       ----------------     --------
       |              |    |       |
      NP              |    V     NP
       |              |    |       |
  -----------         |    |       |
  |        |          |    |       |
NP         |        N-BAR LOVED N-BAR
  |        |          |            |
N-BAR    N-BAR        N            N
  |        |          |            |
  |    --------       |            |
  |    |      |       |            |
  |   ADJP    |       |            |
  |    |      |       |            |
  | ADJ-BAR   |       |            |
  |    |      |       |            |
  N   ADJ   N-BAR  FRIEND      HIMSELF
  |    |      |
 HIS   BIG    N
              |
           BROTHER+S
```

Figure 5. Structured parse tree

```
Input Sentence: his big brother loved himself
Generating a              n -> his
Generating a            adj -> big
Generating a              n -> brother+s
Generating a              n -> friend
Generating a              v -> loved
Generating a              n -> himself

Input Sentence: n adj n n v n
Generating a          n-bar -> n
Generating a        adj-bar -> adj
Generating a          n-bar -> n
Generating a          n-bar -> n
Generating a              v -> v
Generating a          n-bar -> n

Input Sentence: n-bar adj-bar n-bar n-bar v n-bar
Generating a             np -> n-bar
Generating a           adjp -> adj-bar
Generating a          n-bar -> n-bar
Generating a          n-bar -> n-bar
Generating a              v -> v
Generating a             np -> n-bar

Input Sentence: np adjp n-bar n-bar v np
Generating a             np -> np
Generating a          n-bar -> adjp n-bar
Generating a          n-bar -> n-bar
Generating a          v-bar -> v np

Input Sentence: np n-bar n-bar v-bar
Generating a             np -> np n-bar
Generating a          n-bar -> n-bar
Generating a             vp -> v-bar

Input Sentence: np n-bar vp
Generating a             np -> np n-bar
Generating a             vp -> vp

Input Sentence: np v
Generating a              s -> np vp
```

Figure 6. Generation of training data

International Conference on Acoustics, Speech and Signal Processing. School of Computer Science, Carnegie Mellon University, Pitt. PA, USA, 1990.

[5] Ajay N. Jain. *Generalization Performance in Parsec - a Structured Connectionist Parsing Architecture.* In J.E.Moddy, S.J.Hanson, and R.P.Lippman, editors, *Advances in Neural Information Processing Systems 4.* Morgan Kaufmann Pub., 1992.

[6] R.S. Jackendoff. *X-Syntax: A Study of Phrase Structure,* Cambridge Mass: MIT Pres, 1977.

[7] Alex Waibel, Ajay Jain, Arthur McNair, Joe Tebelskis, Louise Osterholtz, Hiroaki Saito, Otto Schmidbauer, Tilo Sloboda, and Monika Woszczyna. *JANUS: Speech-to-Speech Translation Using Connectionist and Non-connectionist Techniques.* In J.E.Moody, S.J.Hanson, and R.P.Lippman, editors, *Advances in Neural Information Processing Systems 4.* Morgan Kaufmann Publishers, 1992.

[8] Louise Osterholtz, Charles Augustine, Arthur McNair, Ivica Rogina, Hiroaki Saito, Tilo Sloboda, Joe Tebelskis, and Alex Waibel. Testing Generality in JANUS: A multi-lingual speech translation system. In *Proceedings of ICASSP.* IEEE, 1992.

[9] R. Mikkulainen and M.G. Dyer *Natural Language Processing with modular PDP Networks and Distributed Lexicon* Cognitive Science, 1991, Vol 15, 343-399.

[10] M.F. St. John *The Story Gestalt: A Model of Knowledge-Intensive Processes in Text Comprehension* Cognitive Science, 1992, Vol 16, 271-306.

[11] T.S. Polzin, *Connectionist Parsing and the Resolution of Anaphoric Relations*, LCL-Technical-Report, Department of Philosophy, Carnegie Mellon University, in preparation.