A CONNECTIONIST MODEL FOR DIALOG PROCESSING

Ye-Yi Wang *Alex Waibel**

School of Computer Science, Carnegie Mellon University
Pittsburgh, Pennsylvania 15213-3890, USA

ABSTRACT

This paper describes a novel connectionist system for dialog processing. Based on a script-like formalism, the system consists of several modular neural networks which can track the semantic flow of a dialog. The system can be extended to understand and translate dialogs in a certain domain.

1. INTRODUCTION

Connectionist models appear to be an attractive framework for natural language processing and high level cognitive processing. Their ability to manipulate symbols through a time sequence makes it possible to process languages; Their ability to integrate multiple constrained sources of information provides an effective formalism for natural language processing, which involves the integration of a variety of linguistic knowledge, such as syntax, semantics and pragmatics.

Recently there has been a lot of work on connectionist models for natural language processing: language acquisition modeling[1][2], connectionist parser construction[3], semantic structure analysis[4][5], etc. A modular connectionist architecture for paraphrasing script-based stories has been described in [6], which combined different neural network modules for various tasks of parsing, semantic analysis and text generation. It has explicit representations for syntactic and semantic structures of sentences.

To process a loosely structured language like spoken language, we propose a more flexible model. In our model there is no predefined syntactic or semantic structure. The model develops by itself a suitable representation of structures and meaning. This enables the model to cope with languages with loose structures effectively. Our system does not have separated syntactic and semantic passes either, which allows us to take full advantage of neural networks' ability to integrate multiple constraints.

The model can track the semantic flow of a dialog in a specific domain, e.g. Conference Registration. With some changes to the basic structure, the model can perform other natural language processing tasks, such as dialog translation.

*This work was partly supported by ATR Interpreting Telephony Research Laboratories, NEC and Siemens.

2. THE MODEL

2.1. The Representation of Words and Sentences

Words are represented in the model as distributed patterns of activation. Each word is hand-coded as a feature pattern in the lexicon. Each bit in the pattern represents a specific syntactic or semantic feature.

Based on the assumption that high level language understanding should consult high level linguistic information other than the meaning of words, we introduce the representation of sentences. Dialog modeling is based on this representation. The sentence representations are also distributed patterns of activation. Since it is relatively difficult to decide which specific features of a sentence are important for dialog understanding, we let the network develop the representations for sentences itself by learning[7]. A learned sentence representation enables the network to achieve better performance.

2.2. The Architecture of the model

The model consists of three components, namely the *Sentence Representation Generator* (SRG) , the *Script Analyst* (SA) and the *Information Extractor* (IE) [Fig. 1]. Each component consists of recurrent neural networks[8].

Sentence Representation Generator (SRG)

Dialog modeling is based on the sentence level representation of sequences of sentences. This raises a problem: how should the meaning of each sentence be represented? It is impractical to manually assign a distributed pattern to each sentence, since there can be an infinite number of sentences. Besides, a manually assigned representation may not best capture the relevant information for dialog level processing.

The SRG is a component which produces the sentence level representation from word feature patterns. Its input is a temporal sequence of feature patterns of words in a sentence, and its output is the representation of the structure and the meaning (the distributed pattern) of that sentence.

At each time t, the pattern for the t th word in a sentence is fed to the SRG as input, along with the activations of the hidden units at time $t-1$ stored in the context (recurrent) units. After all the words in the sentence are fed to the network, the final output of the SRG is the distributed representation of that sentence which encapsulates its syntactic and semantic features.

Script Analyst (SA)

CH2977-7/91/0000-0785 $1.00 © 1991 IEEE

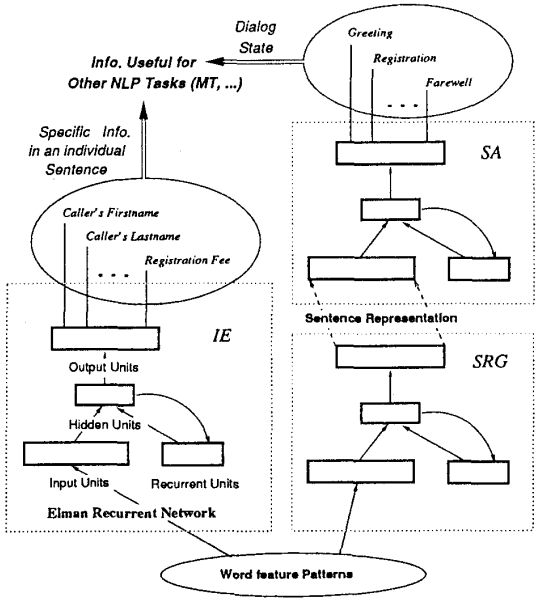

Figure 1: Architecture of the model

The SA fulfills the task of tracking the semantic flow of dialogs. The inputs to the SA are sentence representations, and the outputs from the SA are local representations of the semantic categories (also known as script slots or topics) of the sentences within a dialog.

The SA works in a way similar to the SRG. At time t, it accepts the representation of the t th sentence of a dialog (obtained from the SRG), along with the context information from the hidden layer at time t-1 as inputs; its output specifies the semantic category (or topic) of the input sentence in that context with the corresponding output unit being active. However, in the SA's output representation, only one unit is active. As the topic of the dialog changes, the "state" of the dialog will change and the corresponding output units will flip.

Information Extractor (IE)

In order to understand a dialog, the model should be able to acquire specific information other than the semantic category of a sentence. It should be able to locate and remember the specific information like the caller's name, the application fee, etc., mentioned in the dialog. The third network, the Information Extractor (IE) is introduced for this purpose.

The IE is independent from the other two networks. It takes sequential word patterns as inputs, and activates a specific unit in the output layer if the input word is the corresponding specific information. For example, if the input word is *John*, then the *FirstName* unit is activated in the output layer. The output here is a local representation similar to that of the SA, except that the output here may not have any active unit.

3. THE TRAINING OF THE NETWORKS

The two-leveled architecture (SRG and SA) makes the sentence representation transparent to users, while allowing dialog flow tracking based on high-level semantic information. It is trained in two stages: the training of the SA and the training of the SRG thereafter.

The SA is trained first. Initially each sentence in the training set is represented by a random pattern. An extended backpropagation procedure[7] is then used to modify the weights of the SA *as well as* its input sentence representation. In a sample dialog, the representation of a sentence is fed to the network as input, and all the context units are set to zero initially. A forward pass is performed and the mean square error at the output is calculated. The error is back-propagated, and the weights are modified with gradient descent. The error is however backpropagated one layer further than in the standard backpropagation algorithm to the virtual input layer[7] to modify the weights of the connections between the virtual input layer and the SA's input layer. In this way, the representation of the SA's input pattern is modified during the backward pass. The process is repeated for all sentences in the dialog with the context units always set the activations of the hidden units in the previous step. The network is trained with all the sample dialogs for several epochs until the final output error of each dialog is less than a certain threshold. The final representations of the sentences in the sample dialogs can then be used as the targets for the training of the SRG.

The training for the SRG could be performed in a similar way. At present, however, the input feature patterns of words are already hand-coded, so it is not necessary to train the word representations. We use standard backpropagation to train this network, with the sequence of word feature patterns presented at the input one at a time, and with the pattern of that sentence (developed from the training of the SA) as the target output.

Several measures have been taken during the training to improve the performance of the SRG.

3.1. Sample Balance

One problem which affects the performance of the SRG is that sentences in the sample file are not well balanced: the sentence length varies from 1 word to 21 words. In some case, more than 10 sentences have the same or a similar representation, while some sentences are represented by unique patterns of their own. This makes it hard for the SRG to generate the sentence representation of all the sentences accurately: the network is trained over more backpropagation passes in generating the representations of long sentences or the representations shared by many sentences, whereas the network is inadequately trained to generate an accurate representations of the short sentences or those shared by few sentences. This causes a performance imbalance for the SRG.

To solve this problem, we have normalized the learning rates for different sentences: for long sentences with representations shared by many other sentences, we use small learning rates, for short sentences with representations shared by few sentences, we use large learning rates. The following formula is used to calculate the learning rate for sentence s_i with the representation r_j:

$$\epsilon_i = \frac{\epsilon}{l_i \sqrt{n_j}}$$

Where ϵ is the overall learning rate, ϵ_i is the learning rate for sentence s_i, l_i is the the number of words in s_i, and n_j is the number of sentences with representations similar to r_j.

3.2. Noise and Recurrent Weight Decay

Because of the the limited amount of training data, the SRG may overfit to the initial words in a sentence. For example, in our conference registration task, when the SRG encounters the input words *I, would* and *like*, it will immediately posit that the sentence is *I would like to register for the conference*, and generate the representation of that sentence, since there is no other sentences in the sample file begining with *I would like*. This causes the system to be excessively sensitive to the noise in the begining of a sentence, and makes it hard to train the SRG incrementally for new sentences with the same initial words.

We have taken the following measures to overcome this problem:

- Introducing noise words in the sample sentences: Noise words are randomly chosen from the lexicon, and substituted at random for the actual words of the sentence. No noise words are added to short sentences with fewer than 4 words. The number of noise words is randomly chosen, with the maximum determined by the length of the sentence.

- Decaying weights from context units (recurrent units) to hidden units: In doing so, initial words in a sentence no longer dominate in their effect on the final representation of that sentence. Instead, the SRG generates the representation according to the relevance of the words in a sentence.

4. PERFORMANCE AND EVALUATION

The system is evaluated on the Conference Registration task. The dialogs of this task cover 5 semantic categories, namely *Greeting, Asking for Registration Form, Requesting Caller's Information, Information about Registration Fee* and *Bidding Farewell.* There are about 60 different sentences in the training file. The lexicon consists of 435 words.

The SA network has 30 input units for the sentence representation, 10 context units, 10 hidden units and 5 output units representing the 5 semantic categories. The SRG network consists of 26 input units (each word has 26 features), 16 hidden units, 16 recurrent units and 30 sentence representation output units.

The SA and the SRG are trained separately and then combined to track the semantic flow of dialogs. It is also possible to train the two networks together as a big network for several epochs after each of them has been trained separately. This final tuning might be helpful in improving the performance of the system[9].

The system achieves 100% accuracy on the training set of dialogs and 95% accuracy on the testing set of 15 dialogs containing 246 sentences.

4.1. Ambiguous Sentences

Sentences like "*I understand*", "*Yes, that's right*" and "*Thank you very much*" can occur in almost every category. The networks can however classify these sentences into the correct semantic categories using the context information. For example, if the sentence "*I understand*" follows "*The application fee is 200 dollars per person*", the networks activate the *Information about Registration Fee* dialog output unit. On the other hand, if the sentences is preceded by "*My address is ...*", the networks will activate the *Requesting Caller's Information* unit.

4.2. Complicated Dialog Structure

The model can deal with complicated dialog structures. The speakers may talk about a topic like *registration form* first, then shift to another topic like *application fee*, and then return to the first topic. At each point in the dialog, there may be several possible different successive topics. The system can combine the meaning of sentences together with the context information to choose the correct semantic category for that sentence.

4.3. Generalization and Noise Tolerance

As described above, we use syntactic and semantic feature patterns to represent words, so similar words like "John" and "Bill" have the same feature pattern. This allow the network to gain good generalization performance, so that even if the system did not encounter "*My name is John Smith*" during the training, it can generate the same sentence representation as the sample "*My name is Bill Thompson*".

The system can deal with unfamiliar sentences properly. Because the SRG generates the representation of a sentence in view of the whole sentence pattern rather than a particular word, it is not surprising that it can generate very similar representations for sentences like "*I would like to register for the conference*" and "*I want to register for the conference*". Therefore even though the second sentence is not a training sample, the system can categorize it correctly.

For the same reason, the system can tolerate noise in the input sentence. If the input is "*I would like to register for the registration*" instead of "*I would like to register for the conference*", the SRG will still generate a sentence pattern similar to that of the second sentence.

5. EXPERIMENT IN DIALOG TRANSLATION

The model is extended to a simple dialog translation system. The system determines the meaning of an input sentence, chooses a corresponding target language sentence template, and merges the template and the information obtained from the IE to form the target language sentence [Fig. 2].

On the assumption that the semantic flow of dialogs (in other words, the context for a sentence) is helpful for the translation of that sentence, we embed part of the SA network into the translation network: the connections from the input and context units to the hidden units in the SA are copied to the translation network (the dashed connections in Fig. 2) and kept frozen during the training of the translation network.

However dialog tracking does not equal translation. The sentence representation for tracking semantic flow of dialogs provides insufficient information for dialog translation. Because the sentence representation for the SA is developed automatically during the training pass of the SA, all the information it contains is only

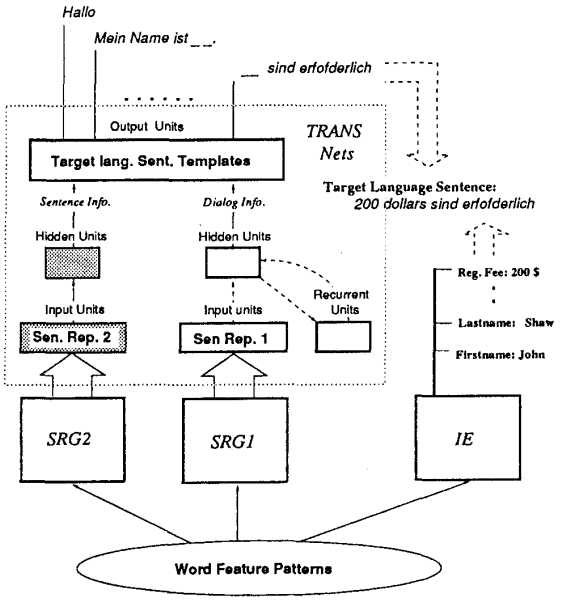

Figure 2: Architecture of the translation networks

sufficient for dialog tracking. Two sentences which occur typically in the same dialog category are usually represented by similar sentence patterns and it becomes impossible to discriminate between them.

To overcome this problem, we introduce another sentence representation as a complement to the previous one to distinguish the different sentences in the same semantic category of a dialog. This additional module uses 20 bit patterns to represent sentences, which can be obtained from another SRG, and is fed to the translation network as input (the shadowed part in Fig. 2).

The development of sentence representation 2 is similar to that of sentence representation 1: initially it is randomly generated for the sentences input to the translation network. During the training of the translation network, it is modified with extended back propagation. The final representation is used as target for the training of the SRG2.

With the same test dialogs as those for dialog state tracking, the networks have been able to translate the sentences with 98% accuracy in semantic sense. This accuracy is higher than that of dialog tracking. This is because that the additional sentence representation helps to remove the errors in dialog tracking.

6. SUMMARY

The connectionist model for dialog processing described here can track semantic flow and perform simple dialog translation efficiently as demonstrated in a domain of conference registration dialogs. In this model, words and sentences are represented as distributed patterns of activation. The representations for sentences are "hidden" and emerge during training with extended backpropagation. A modular network architecture is used, with each module fulfilling a sub-task of language understanding.

In contrast with other more conventional models for language processing, this model automatically "learns" from examples and has no separate syntactic and semantic passes. Instead, it treats syntax and semantics as integrated constraints for the same cognitive tasks. In doing so, it avoids explicit hand coded representations of syntactic and semantic structures. These advantages might be particularly useful in dealing with the variability and noise typical for ill-formed automatically recognized spoken dialog utterances.

REFERENCES

[1] A. L. Gorin and S. E. Levinson. Adaptive acquisition of language. Technical report, Speech Research Department, AT&T Bell Laboratories, Murray Hill, 1989.

[2] D. E. Rumelhart and J. L. McClelland. On learning the past tense of English verbs. In J. L. McClelland and D. E. Rumelhart, editors, *Parallel Distributed Processing*. The MIT Press, 1986.

[3] A. N. Jain and A. H. Waibel. A connectionist parser aimed at spoken language. In *Proceedings of the International Workshop on Parsing Technologies*, Pittsburgh, PA, August 1989. Carnegie Mellon University.

[4] J. L. McClelland and A. H. Kawamoto. Mechanisms of sentence processing: Assigning roles to constituents. In J. L. McClelland and D. E. Rumelhart, editors, *Parallel Distributed Processing*. The MIT Press, 1986.

[5] D. Waltz and J. Pollack. Massively parallel parsing: A strongly interactive model of natural language interpretation. *Cognitive Science*, 9, 1985.

[6] R. Miikkulainen and M. G. Dyer. A modular neural network architecture for sequential paraphrasing of script-based stories. In *Proceedings of the International Joint Conference on Neural Networks*. IEEE, 1989.

[7] R. Miikkulainen and M. G. Dyer. Encoding input/output representations in connectionist cognitive systems. In *Proceedings of the 1988 Connectionist Models Summer School*. Morgan Kaufmann Publishers, 1989.

[8] J. L. Elman. Finding structure in time. Technical Report 8801, Center for Research in Language, University of California, San Diego, 1988.

[9] A. H. Waibel, H. Sawai, and K. Shikano. Modularity and scaling in large phonemic neural networks. Technical Report ATR-I-0034, ATR Interpreting Telephony Research Laboratories, 1988.