Neural Probabilistic Structured Language Modeling

Ahmad Emami and Frederick Jelinek
CLSP/Electrical and Computer Engineering

AIMS We investigate the performance of the Structured Language Model when one of its components is modeled by a connectionist model. Using a connectionist model and a distributed representation of the items in the history allows the component to use much longer contexts than possible with current interpolated or back-off models, both because of the inherent capability of the connectionist model to fight the data sparseness problem, and because of the sub-linear growth in the model size when increasing the context length.

INTRODUCTION N-gram language models are the most commonly used models in speech recognition systems. Despite their naive underlying assumption, N-gram models perform surprisingly well. However they suffer from severe data sparseness, and they are intrinsically unable to use long contexts for prediction.

In the Structured Language Model (SLM) [1], long contexts are used for prediction by means of building partial syntactical parses on the prefix word strings and using information extracted from these partial parses.

There has been promising work in using distributional representation of words and neural networks for language modeling [2]. One great advantage of this approach is its ability to fight data sparseness. The model size grows only sub-linearly with the number of predicting features used. It has been shown that this method improves on regular N-gram models in perplexity [2].

In this work we investigate the impact of using a neural network model as the predictor component of the Structured Language Model, giving it the ability to use many more features than the original SLM while avoiding data sparseness.

STRUCTURED LANGUAGE MODEL An extensive presentation of the SLM can be found in [1]. The model assigns a probability $P(W,T)$ to every sentence W and every possible binary parse T of W. The terminals of T are the words of W with POS tags, and the nodes of T are annotated with phrase headwords and non-terminal labels.

The SLM is made of three components, starting with the PREDICTOR predicting the next word; then moving on to the TAGGER to tag the newly predicted word; and then using the CONSTRUCTOR to build partial parses for the newly extended word string; after the CONSTRUCTOR the model passes control to the PREDICTOR again.

The *language model* probability assignment for the word at position $k+1$ in the input sentence is made using:

$$P_{SLM}(w_{k+1}|W_k) = \sum_{T_k \in S_k} P(w_{k+1}|W_k T_k) \cdot \rho(W_k, T_k), \quad (1)$$

$$\rho(W_k, T_k) = P(W_k T_k) | \sum_{T_k \in S_k} P(W_k T_k), \quad (2)$$

which ensures a proper probability normalization over strings W^*, where S_k is the set of all parses present in our stacks at the current word string depth k.

NEURAL NETWORK MODEL In brief, this model can be described as follows: a *feature vector* is associated with each token in some given *input vocabulary*. The input to the neural network is then composed of a single vector which is the concatenation of the feature vectors of the items in the history. The neural network then computes the (conditional) distribution over all tokens in the *output vocabulary*, given the input described above.

Training is achieved by searching for parameters Φ of the neural network and the values of feature vectors that maximize the penalized log-likelihood of the training corpus:

$$L = \frac{1}{T} \sum_t log P(y^t | x_1^t, ..., x_{n-1}^t; \Phi) - R(\Phi) \quad (3)$$

where $P(y^t|x_1^t, ..., x_{n-1}^t)$ denotes the conditional probability of word y^t (x_i and y belonging to the input and output vocabularies V_i and V_o respectively) at word position t, given its history $x_1^t, ..., x_{n-1}^t$. T is the size of the training data and $R(\Phi)$ is a regularization term, second norm of the parameters in our case.

		+slm	+3gm	+5gm
SLM	161	161	137	132
2HW	174	137	127	123
3HW	161	132	123	119
HW-OP	155	129	121	**117**

Table 1: UPENN section perplexity

		+slm	+lattice	+5gm	+l&5gm
Lattice	13.7	12.6	13.7	13.2	13.2
SLM	12.7	12.7	12.6	12.7	12.6
2HW	13.5	12.7	12.7	12.5	**12.4**
3HW	13.6	12.6	12.8	12.7	12.6
HW-OP	13.2	12.5	12.9	**12.4**	**12.4**

Table 2: WSJ word error rate

NEURAL NETWORK IN SLM We used a neural net model as one of the two PREDICTOR components of the Structured Language Model. The other components of the SLM remain unchanged, the reasons being the rather high computational complexity of neural nets, and the fact that the PREDICTOR is the component most affected by data sparseness. Furthermore, a neural net was used only as the PREDICTOR for the language modeling (Equation 1) part of the SLM, not for the building of partial parses.

EXPERIMENTS Table 1 gives the perplexity results on the UPENN section of the Wall Street Journal (WSJ) corpus. The vocabulary size was 10,000 and there were a total of 94 non-terminal tags and part of speech tags. The rows denoted by 2HW, 3HW, and HW-OP correspond to contexts consisting of 2 previous heads, 3 previous heads, and 3 previous heads plus the first previous opposite head. The $n-th$ previous opposite head is the child of the $n-th$ previous head that is not the head itself. The columns +slm, +3gm, and +5gm denote linear interpolation with the baseline SLM, a 3-gram back-off, and 5-gram back-off models respectively. The baseline SLM is the same as our connectionist based one, except that it uses an interpolated N-gram model for the PREDICTOR in its language model part.

Similarly, Table 2 gives the Word Error Rate (WER) results on re-ranking the output of a speech recognizer on the Wall Street Journal Corpus. The output vocabulary was limited to the 5,000 most frequent words of the original 19,006 words vocabulary, with the scores of the other words replaced by those obtained from a 5-gram model. Here the row lattice denotes the language model from the speech recognizer producing the N-best list and the column +l&5gm denotes interpolation with the lattice and the back-off 5-gram models.

FUTURE WORK In this work the neural network model was used as the PREDICTOR for the language model part of the SLM, trained on single Gold Standard parses (produced either by humans or by reliable outside parser). We can alternatively train the neural network on the partial parses constructed by the SLM itself.

Furthermore, the neural network model can be used for all the components of the SLM. Another enhancement may come from full embedded training of a neural network based SLM, initializing the model, building partial parses, and re-training again on the obtained partial parse; iterating the procedure above till convergence is achieved.

Publications

1. Ahmad Emami, Peng Xu and Frederick Jelinek, "Using a connectionist model in a syntactical based language model", Proc. ICASSP, 2003 (accepted)
2. Ahmad Emami, "Improving a Connectionist Based Syntactical Language Model", Proc. Eurospeech 2003 (submitted)

References

[1] C. Chelba and F. Jelinek, "Structured language modeling," *Computer Speech and Language*, vol. 14, pp. 283–332, October 2000.

[2] Y. Bengio, R. Ducharme, and P. Vincent, "A neural probabilistic language model," in *Advances in Neural Information Processing Systems*, 2001.