Improve the Performance of Neural Network Lexicon Model

Jiahui Geng

jgeng@cs.rwth-aachen.de

November 3, 2017

Human Language Technology and Pattern Recognition
Lehrstuhl für Informatik 6
Computer Science Department
RWTH Aachen University, Germany

Introduction

Motivation: Improve the performance of neural lexicon model with more context

- Worse than SMT in low-resource scenarios [Koehn & Knowles 17]
- How can we exploit monolingual data in NMT?

This talk:

- Literature review
- Which of them are useful for evaluation campaigns?
  - Low-resource scenarios, e.g. English-Romanian
  - Resource-rich scenarios, e.g. German-English
- Extension of ideas

Introduction

Information

- Source: embedding, reordering, adequacy
- Target: lm, fluency

Usage

- Generate parallel data
- Train only with monolingual data
- Extend model architecture

Outline

Hybrid ANN Approach

We train the ANN lexicon models using maximum likelihood estimation. Let x_1^N be the training data (concatenation of all training sentences including sentence start/end tokens). θ be a set of lexicon parameters to learn. The training criterion (discriminative non-context case) is given below:

$$\begin{aligned} &\operatorname*{argmax}_{\theta} p(x_1^N; \theta) \\ = &\operatorname*{argmax}_{\theta} \sum_{c_1^N} p(x_1^N, c_1^N; \theta) \end{aligned} \tag{1}$$

Plugging this into the auxiliary objective of the EM algorithm yields a crossentropy-like function. We hope to maximize function Q in EM iterations.

Basic Math Formula

$$
\begin{aligned}
&Q(\hat{\theta}, \theta) \\
&= \sum_{c_1^N} p(c_1^N | x_1^N; \theta) \cdot \log\, p(c_1^N, x_1^N; \hat{\theta}) \\
&\approx \sum_{c_1^N} p(c_1^N | x_1^N; \theta) \cdot \sum_n \log\, \frac{p(c_n | x_n; \hat{\theta})}{p(c_n)} \\
&\approx \sum_n \sum_{c_1^N} p(c_1^N | x_1^N; \theta) \cdot \log\, p(c_n | x_n; \hat{\theta}) \\
&= \sum_n \sum_c \sum_{c_1^N : c_n = c} p(c_1^N | x_1^N; \theta) \cdot \log\, p(c | x_n; \hat{\theta}) \\
&= \sum_n \sum_c p_n(c | x_1^N; \theta) \cdot \log\, p(c | x_n; \hat{\theta})
\end{aligned}
\tag{2}
$$

And ideally we expect better results by replacing $p(c|x_n)$ with $p(c|x_n, x_{n-1})$, $p(c|x_{n+1}, x_n, x_{n-1})$ which contains more context information.

Experiments with different number of context characters

Comparison with the Pretrained Neural Network

Pretraind the neural lexicon network with a small parallel dataset. We may find that the optimum value achieved by our model is near the value from a pre_trained neural lexicon network. That should be the optimum value of our unsupervised method.

Problems in

The final MSER increase when the number of context characters increase.

In order to make our method more general for lexicon model with more context characters. Improvement proposals:

Redesign the output layer including the softmax function, we hope to get a different or even better optimum result.
Reformulate the mathematical formula to generate model suitable for more context character cases.

Quadratic Softmax

The original optimization criterion:

$$\begin{aligned}
&\operatorname*{argmax}_{\theta} \sum_{c_1^N} p(x_1^N, c_1^N; \theta) \\
=&\operatorname*{argmax}_{\theta} \{ \sum_{c_1^N} q(c_1^N) \cdot \prod_n q(x_n | c_n) \} \\
=&\operatorname*{argmax}_{\theta} \{ \sum_{c_1^N} q(c_1^N) \cdot \prod_n \frac{q(c_n | x_n)}{q_n(c_n)} \} \\
=&\operatorname*{argmax}_{\theta} \{ \sum_{c_1^N} \frac{q(c_1^N)}{\prod_n q_n(c_n)} \cdot \prod_n q_n(c_n | x_n) \}
\end{aligned}$$

distance interpretation:

$$\sum_{c_1^N} \left(\frac{q(c_1^N)}{\prod_n q_n(c_n)} - \prod_n q_n(c_n | x_n) \right)^2$$

Quadratic Softmax

Expand the formula we get:

$$\sum_{c_1^N} \frac{(q^2(c_1^N))}{\prod_n q_n^2(c_n)} - 2 \cdot \sum_{c_1^N} q(c_1^N) \cdot \prod_n \frac{q_n(c_n|x_n)}{q_n(c_n)} + \sum_{c_1^N} \prod_n q_n^2(c_n|x_n)$$

minimization of distance would be equivalent for quadratic absolute normalization:

$$\sum_c q_n^2(c|x_1^N) = 1.0$$

Implementation:

$$softmax \Rightarrow \frac{e^{y_c}}{\sum_c e^{y_c}}$$

$$quadratic\ softmax \Rightarrow \sqrt{\frac{e^{y_c}}{\sum_c e^{y_c}}}$$

Prior Softmax

Another distance interpretation:

$$\sum_{c_1^N} (q(c_1^N) - \prod_n \frac{q(c_n|x_n)}{q_n(c_n)})^2$$

Unfold the formula we get:

$$\sum_{c_1^N} q^2(c_1^N) - 2 \cdot \sum_{c_1^N} q(c_1^N) \cdot \prod_n \frac{q_n(c_n|x_n)}{q_n(c_n)} + \sum_{c_1^N} \prod_n \frac{q_n^2(c_n|x_n)}{q_n^2(c_n)}$$

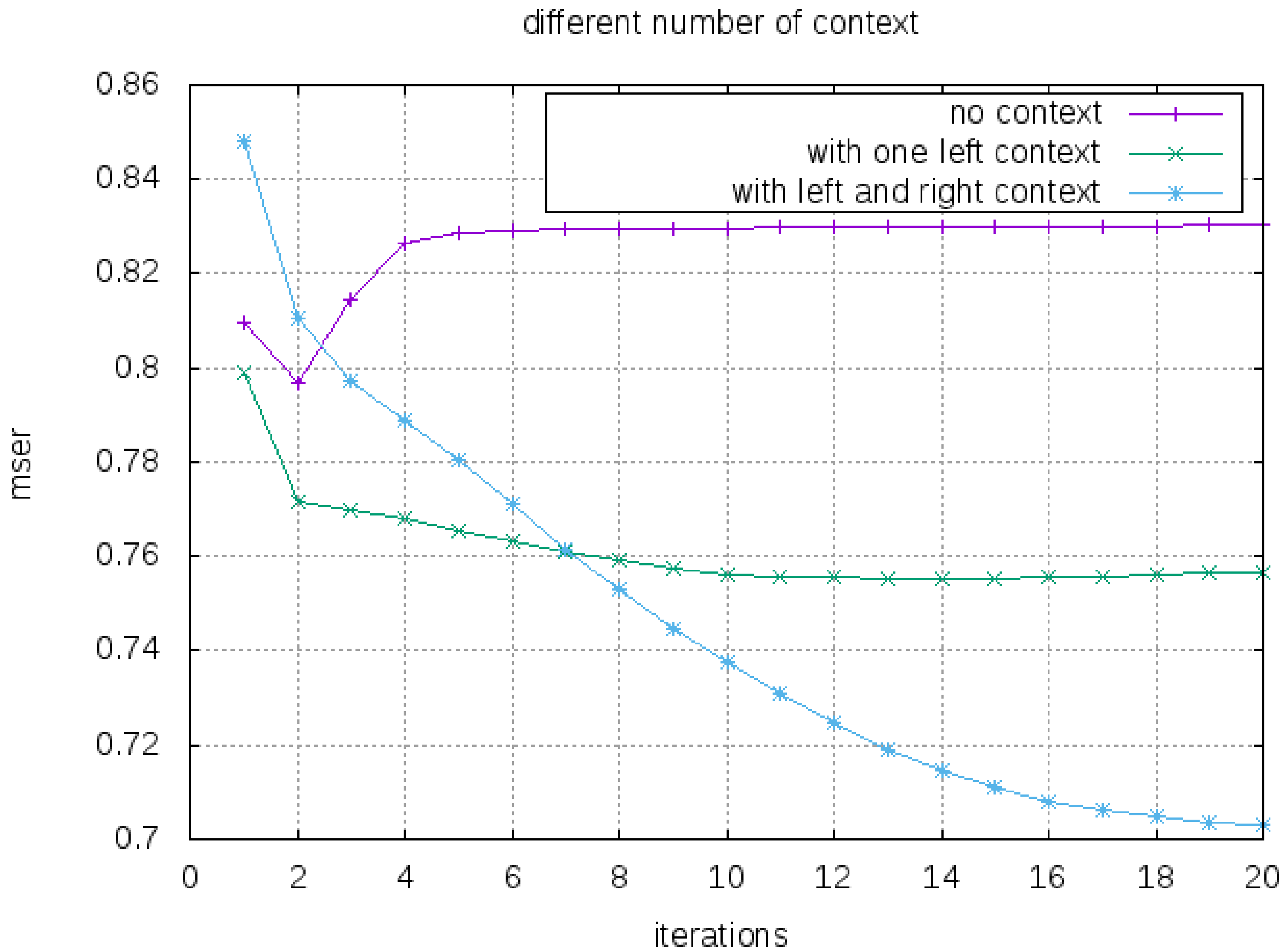
The constraint:

$$\sum_c \frac{q_n^2(c|x_n)}{q_n^2(c)} = 1.0$$

Implementation:

$$prior\ softmax \Rightarrow \sqrt{\frac{e^{y_c}}{\sum_c e^{y_c}}} \cdot p(c)$$

Prior Softmax Experiments



Conclusion and Outlook

Conclusion **method A seems the most promising**

Outlook **systematic comparison on a common task apply to next evaluation campaigns develop new ideas**

Thank you for your attention

Jiahui Geng

jgeng@cs.rwth-aachen.de

http://www.hltpr.rwth-aachen.de/

References

[Bahdanau & Cho^{+} 15] D. Bahdanau, K. Cho, Y. Bengio: Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, CA, USA, May 2015.

[Koehn & Knowles 17] P. Koehn, R. Knowles: Six Challenges for Neural Machine Translation. In *Proceedings of the ACL 2017 1st Workshop on Neural Machine Translation (NMT 2017)*, Vancouver, Canada, August 2017.