Neural Hidden Markov Model for Machine Translation

Weiyue Wang, Derui Zhu, Tamer Alkhouli, Zixuan Gan and Hermann Ney

{surname}@i6.informatik.rwth-aachen.de

July 17th, 2018

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

► Attention-based neural translation models
  ▶ attend to specific positions on the source side to generate translation
  ▶ improvements over pure encoder-decoder sequence-to-sequence approach

► Neural HMM has been successfully applied on top of SMT systems [Wang & Alkhouli\textsuperscript{+} 17]

► This work explores its application in standalone decoding
  ▶ end-to-end, only with neural networks $\rightarrow$ NMT
  ▶ LSTM structures outperform FFNN variants in [Wang & Alkhouli\textsuperscript{+} 17]
Neural Hidden Markov Model

Translation

- source sentence $f_1^J = f_1 ... f_j ... f_J$
- target sentence $e_1^I = e_1 ... e_i ... e_I$
- alignment $i \rightarrow j = b_i$

Model translation using an alignment model and a lexicon model:

$$p(e_1^I | f_1^J) = \sum_{b_1^I} p(e_1^I, b_1^I | f_1^J)$$

(1)

$$:= \sum_{b_1^I} \prod_{i=1}^{I} \underbrace{p(e_i | b_1^i, e_0^{i-1}, f_1^J)}_{\text{lexicon model}} \cdot \underbrace{p(b_i | b_1^{i-1}, e_0^{i-1}, f_1^J)}_{\text{alignment model}}$$

(2)

with $p(b_i | b_1^{i-1}, e_0^{i-1}, f_1^J) := p(\Delta_i | b_1^{i-1}, e_0^{i-1}, f_1^J)$

- predicts the jump $\Delta_i = b_i - b_{i-1}$
Neural Hidden Markov Model

▶ Neural network based lexicon model
Neural Hidden Markov Model

▶ Neural network based alignment model \((j' = b_{i-1})\)
Training

Training criterion for sentence pairs \((F_r, E_r), r = 1, \ldots, R:\)

\[
\arg\max_{\theta} \left\{ \sum_r \log p_{\theta}(E_r | F_r) \right\}
\]  

(3)

Derivative for a single sentence pair \((F, E) = (f_1^J, e_1^I):\)

\[
\frac{\partial}{\partial \theta} \log p_{\theta}(E | F) = \sum_{j', j} \sum_i p_i(j', j| f_1^J, e_1^I; \theta) \cdot \frac{\partial}{\partial \theta} \log p(j, e_i | j', e_i^{-1}, f_1^J; \theta)
\]  

(4)

Entire training procedure: backpropagation in an EM framework

1. compute:
   - the HMM posterior weights
   - the local gradients (backpropagation)

2. update neural network weights
Decoding

Search over all possible target strings

\[
\max_{e^I_1} p(e^I_1|f^J_1) = \max_{e^I_1} \left\{ \sum_{b^I_i} \prod_{i} p(b_i, e_i|b_{i-1}, e^{i-1}_0, f^J_1) \right\}
\]

Extending partial hypothesis from \(e^{i-1}_0\) to \(e^i_0\)

\[
Q(i, j; e^i_0) = \sum_{j'} [p(j, e_i|j', e^{i-1}_0, f^J_1) \cdot Q(i - 1, j'; e^{i-1}_0)]
\]  \(5\)

Pruning:

\[
Q(i; e^i_0) = \sum_{j} Q(i, j; e^i_0)
\]

\[
\text{argmax}_{e^i_0} Q(i; e^i_0) \leftarrow \text{select several candidates}
\]  \(6\)
Decoding

- No explicit coverage constraints
  - one-to-many alignment cases and unaligned source words
- Search space in decoding
  - neural HMM: consists of both alignment and translation decisions
  - attention model: consists only of translation decisions
- Decoding complexity \((J = \text{source sentence length}, I = \text{target sentence length})\)
  - neural HMM: \(O(J^2 \cdot I)\)
  - attention model: \(O(J \cdot I)\)
  - in practice, neural HMM 3 times slower than attention model
Experimental Setup

- WMT 2017 German↔English and Chinese→English translation tasks
- Quality measured with case sensitive BLEU and TER on newstests2017
- Moses tokenizer and truecasing scripts [Koehn & Hoang+ 07]
- Jieba¹ segmenter for Chinese data
- 20K byte pair encoding (BPE) operations [Sennrich & Haddow+ 16]
  - joint for German↔English and separate for Chinese→English
- Attention-based system are trained with Sockeye [Hieber & Domhan+ 17]
  - encoder and decoder embedding layer size 620
  - a bidirectional encoder layer with 1000 LSTMs with peephole connections
  - Adam [Kingma & Ba 15] as optimizer with a learning rate of 0.001
  - batch size 50, 30% dropout
  - beam search with beam size 12
  - model weights averaging

¹https://github.com/fxsjy/jieba
Experimental Setup

- Neural hidden markov model implemented in *TensorFlow* [Abadi & Agarwal 2016]
  - encoder and decoder embedding layer size 350
  - projection layer size 800 (400+200+200)
  - three hidden layers of sizes 1000, 1000 and 500 respectively
  - normal softmax layer
    - lexicon model: large output layer with roughly 25K nodes
    - alignment model: small output layer with 201 nodes
  - *Adam* as optimizer with a learning rate of 0.001
  - batch size 20, 30% dropout
  - beam search with beam size 12
  - model weights averaging
## Experimental Results

<table>
<thead>
<tr>
<th>WMT 2017</th>
<th># free parameters</th>
<th>German $\rightarrow$ English</th>
<th>English $\rightarrow$ German</th>
<th>Chinese $\rightarrow$ English</th>
</tr>
</thead>
<tbody>
<tr>
<td>FFNN-based neural HMM</td>
<td>33M</td>
<td>28.3 51.4</td>
<td>23.4 58.8</td>
<td>19.3 64.8</td>
</tr>
<tr>
<td>LSTM-based neural HMM</td>
<td>52M</td>
<td>29.6 50.5</td>
<td>24.6 57.0</td>
<td>20.2 63.7</td>
</tr>
<tr>
<td>Attention-based network</td>
<td>77M</td>
<td>29.5 50.8</td>
<td>24.7 57.4</td>
<td>20.2 63.8</td>
</tr>
</tbody>
</table>

- **FFNN-based neural HMM**: [Wang & Alkhouli $^+$ 17] applied in decoding
- **LSTM-based neural HMM**: this work
- **Attention-based neural network**: [Bahdanau & Cho $^+$ 15]
- All models trained without synthetic data
- Single model used for decoding

- LSTM models improve FFNN-based system by up to 1.3% **BLEU** and 1.8% **TER**
- Comparable performance with attention-based system
Summary

► Apply NNs to conventional HMM for MT
► End-to-end with a stand-alone decoder
► Comparable performance with the standard attention-based system
  ▶ significantly outperforms the feed-forward variant

► Future work
  ▶ Speed up training and decoding
  ▶ Application in automatic post editing
  ▶ Combination with attention or transformer [Vaswani & Shazeer 2017] model
Thank you for your attention

Weiyue Wang

wwang@cs.rwth-aachen.de

http://www-i6.informatik.rwth-aachen.de/
Appendix: Motivation

► Neural HMM compared to attention-based systems
  ▶ recurrent encoder and decoder without attention component
  ▶ replacing attention mechanism by a first-order HMM alignment model
    ◦ attention levels: deterministic normalized similarity scores
    ◦ HMM alignments: discrete random variables and must be marginalized
  ▶ separating the alignment model from the lexicon model
    ◦ more flexibility in modeling and training
    ◦ avoids propagating errors from one model to another
    ◦ implies an extended degree of interpretability and control over the model
Appendix: Analysis

Attention weight and alignment matrices visualized in heat map form

Generated by attention NMT baseline and neural HMM
## Appendix: Analysis

<table>
<thead>
<tr>
<th></th>
<th>source</th>
<th>reference</th>
<th>attention NMT</th>
<th>neural HMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>28-jähriger Koch in San Francisco Mall tot aufgefunden</td>
<td>28-Year-Old Chef Found Dead at San Francisco Mall</td>
<td>28-year-old cook in San Francisco Mall found dead</td>
<td>28-year-old cook found dead in San Francisco Mall</td>
</tr>
<tr>
<td>2</td>
<td>Frankie hat in GB bereits fast 30 Jahre Gewinner geritten, was toll ist.</td>
<td>Frankie ‘s been <em>riding winners</em> in the UK for the best part of 30 years which is great to see.</td>
<td>Frankie has been a <em>winner</em> in the UK for almost 30 years, which is great.</td>
<td>Frankie has <em>ridden winners</em> in the UK for almost 30 years, which is great.</td>
</tr>
<tr>
<td>3</td>
<td>Wer baut Braunschweigs günstige Wohnungen?</td>
<td>Who is going to <em>build</em> Braunschweig ‘s low-cost housing?</td>
<td>Who <em>does</em> Braunschweig <em>build</em> cheap apartments?</td>
<td>Who <em>builds</em> Braunschweig ‘s cheap apartments?</td>
</tr>
</tbody>
</table>

▶ Sample translations from the WMT German→English newstest2017 set

- **underline** source words of interest
- **italicize** *correct* translations
- **bold-face** for *incorrect* translations
References


