Improving Unsupervised Word-by-Word Translation Using Language Model and Denoising Autoencoder

Yunsu Kim, Jiahui Geng, Hermann Ney
kim@cs.rwth-aachen.de

EMNLP 2018
November 2, 2018

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

Machine translation (MT) requires lots of parallel data
  ► Especially for neural models [Koehn & Knowles 17]
  ► Small or no parallel data for many language pairs

Unsupervised MT: Train only with monolingual data
  ► [Artetxe & Labaka\textsuperscript{+} 18], [Lample & Denoyer\textsuperscript{+} 18]
  ► Iterative back-translation of both translation directions
    ▶ Long training time (e.g. 1-3 weeks)
  ► Model shared for both translation directions but separate training data
    ▶ Considerable effort to implement

Can we build an unsupervised machine translation system quickly & simply?
Our Unsupervised MT System

\[ q(e|f) \]

Cross-lingual Word Embedding

\[ f^J_1 \rightarrow + \rightarrow \tilde{e}^J_1 \]

\[ p(e_j|e^j_{1-1}) \]

Language Model

\[ \tilde{e}^J_1 \rightarrow p(e^I_1|\tilde{e}^J_1) \]

Denoising Autoencoder

\[ e^I_1 \]

Combine the ideas from

- Classic word-based models
- Modern neural sequence-to-sequence model

Minimal implementation & Quick training (1-2 days)

- Outperforms [Artetxe & Labaka\textsuperscript{+ 18}], [Lample & Denoyer\textsuperscript{+ 18}]
Word Lexicon: Cross-lingual Word Embedding

Monolingual word embedding
- Skip-gram, CBOW
- Individually learned for source and target

Cross-lingual word embedding
- Linear mapping: source $\rightarrow$ target
- Shared embedding space
- Arithmetic operations possible between source and target words
Word Lexicon: Cross-lingual Word Embedding

Unsupervised learning of cross-lingual mapping

1. Initialization: adversarial training [Conneau & Lample+ 18]
2. Training: minimum squared error (MSE)

\[ \hat{W} = \arg\min_{W} \left\{ \sum_{(f,e) \in D} \| W f^{emb} - e^{emb} \| \right\} \]

Dictionary \( D \): mutual nearest neighbors

3. Repeat dictionary induction and MSE training [Artetxe & Labaka+ 17]

Word translation = Nearest neighbor search

\[ \hat{e}(f) = \arg\min_{e} \{ d(f, e) \} \]

\( d(f, e) \): cosine similarity with hub penalty [Conneau & Lample+ 18]
Beam Search with Language Model

Word-by-word translation does not consider **context**

- And most literature on cross-lingual word embedding evaluate only on word translations!
- Ignored so far: behavior of cross-lingual neighbor words within a context

Beam search with language model (LM)

\[
S(e; f, h) = \lambda_{\text{emb}} \log q(f, e) + \lambda_{\text{LM}} \log p(e|h)
\]

- \(q(f, e) \in [0, 1] \): **linearly scaled cosine similarity**
- \(e = k\)-nearest neighbors
- Context-aware lexical choices
Denoising Autoencoder

Cross-lingual word embedding + LM = $f^J_I \rightarrow \tilde{e}^J_I$

- Still one target word per source word
- Reordering is not considered

Denoising: noisy target sentence $\rightarrow$ clean target sentence

- Neural sequence-to-sequence autoencoder
- Can be trained only with target monolingual data

$$L(E) = - \sum_{e^I_1 \in E} \log p(e^I_1 | \text{noise}(e^I_1))$$

- Input noise($e^I_1$): target sentence with artificial noise
  - Simulate errors in word-by-word translations
- Output $e^I_1$: target sentence (original)
Insertion Noise

Case 1: multiple source words → a single target word

Insertion noise: insert a word between original words [This work]
- Randomly with a probability $p_{\text{ins}}$ at each position
- Only $V_{\text{ins}}$ frequent words are inserted, e.g. articles, prepositions

Denoiser learns to delete such words
Deletion Noise

Case 2: a single source word → multiple target words

Deletion noise: delete words from the original sentence [Hill & Cho 16]
- randomly with a probability $p_{\text{del}}$ at each position
- Denoiser learns to insert such words
Case 3: target hypothesis words should be reordered

Permutation noise: permute original word positions [Hill & Cho+ 16]
- randomly within a limited distance $d_{\text{per}}$: maintain general monotonicity
- Denoiser learns to reorder such words
Experimental Setup

Training data: WMT News Crawl monolingual data

- **English**: 100M sentences
- **German**: 100M sentences
- **French**: 42M sentences

Test sets: WMT News translation task

- **German**↔**English**: newstest2016
- **French**↔**English**: newstest2014
Experimental Setup

Cross-lingual word embedding
  ▶ Discriminator input and dictionary induction: 100k frequent words

LM: 5-gram with modified Kneser-Ney smoothing

Denoising autoencoder: 6-layer Transformer encoder/decoder
  ▶ 50k frequent words + <unk>

Search parameters
  ▶ Number of nearest neighbors ($k$) = 100
  ▶ Beam size = 10
  ▶ $\lambda_{\text{emb}} = 1.0$, $\lambda_{\text{LM}} = 0.1$
### BLEU [%] scores on WMT tasks

<table>
<thead>
<tr>
<th>System</th>
<th>de-en</th>
<th>en-de</th>
<th>fr-en</th>
<th>en-fr</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>newstest2016</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>newstest2014</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Word-by-Word</td>
<td>11.1</td>
<td>6.7</td>
<td>10.6</td>
<td>7.8</td>
</tr>
<tr>
<td>+ LM</td>
<td>14.5</td>
<td>9.9</td>
<td>13.6</td>
<td>10.9</td>
</tr>
<tr>
<td>+ Denoising</td>
<td>17.2</td>
<td>11.0</td>
<td>16.5</td>
<td>13.9</td>
</tr>
<tr>
<td>[Lample &amp; Denoyer(^+) 18]</td>
<td>13.3</td>
<td>9.6</td>
<td>14.3</td>
<td>15.1</td>
</tr>
<tr>
<td>[Artetxe &amp; Labaka(^+) 18]</td>
<td>-</td>
<td>-</td>
<td>15.6</td>
<td>15.1</td>
</tr>
</tbody>
</table>
Conclusion

Fully unsupervised MT system with cross-lingual word embedding

- Beam search with LM for context-aware lexicon choice
- Denoising autoencoder for insertion/deletion/local reordering
- **Simple** to implement and **fast** to train
- **Outperforms** unsupervised neural MT with iterative back-translations

Future work

- **Our method to initialize unsupervised neural MT**
  [Lample & Denoyer\textsuperscript{+} 18, Artetxe & Labaka\textsuperscript{+} 18]
- Artificial noises to regularize neural MT

Codes available at [https://github.com/yunsukim86/wbw-lm/](https://github.com/yunsukim86/wbw-lm/)
Thank you for your attention

Yunsu Kim

kim@cs.rwth-aachen.de

http://www.hltpr.rwth-aachen.de/
### Ablation Study: Denoising

**$d_{\text{per}}$:** local reordering range / **$p_{\text{del}}$:** deletion probability / **$p_{\text{ins}}$:** insertion vocabulary size

<table>
<thead>
<tr>
<th>$d_{\text{per}}$</th>
<th>$p_{\text{del}}$</th>
<th>$V_{\text{ins}}$</th>
<th>BLEU [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>14.7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>14.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>14.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.1</td>
<td>15.7</td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>15.1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>16.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>17.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>16.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>16.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Ablation Study: Vocabulary

<table>
<thead>
<tr>
<th>Vocabulary</th>
<th>BLEU [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Merges</strong></td>
<td></td>
</tr>
<tr>
<td>20k</td>
<td>10.4</td>
</tr>
<tr>
<td>BPE</td>
<td></td>
</tr>
<tr>
<td>50k</td>
<td>12.5</td>
</tr>
<tr>
<td>100k</td>
<td>13.0</td>
</tr>
<tr>
<td><strong>Cross-lingual training</strong></td>
<td></td>
</tr>
<tr>
<td>20k</td>
<td>14.4</td>
</tr>
<tr>
<td>Word</td>
<td></td>
</tr>
<tr>
<td>50k</td>
<td>14.4</td>
</tr>
<tr>
<td>100k</td>
<td><strong>14.5</strong></td>
</tr>
<tr>
<td>200k</td>
<td>14.4</td>
</tr>
</tbody>
</table>

- Word embedding performs better than BPE embedding
- Embedding trained on 20k similar to 200k $\Rightarrow$ Frequent words matter
References


