Unsupervised Tokenization for Machine Translation

Tagyoung Chung    Daniel Gildea

Computer Science Department
University of Rochester
Tokenization

• Usually the first step of the SMT

• Two different problems for different languages
  – Finding unknown word boundaries
    (isolating languages such as Chinese)
  – Finding morpheme groupings of right granularity
    (agglutinative languages such as Korean)

• Supervised methods require training data or set of rules

• We present two unsupervised methods for MT
Problem of Granularity - Examples

• Korean word *meok-eoss-da* consists of three morphemes
  – *eat-past-indicative* (*TRANSLATION: ate*)
  – No reason to separate morphemes

• Korean word *hakgyo-e* consists of two morphemes
  – *school-locative* (*TRANSLATION: at school*)
  – It is desirable to separate morphemes in this case

• Bilingual model may help with this issue
Overview

• Monolingual model
  – Model description
  – Handling overfitting

• Bilingual model
  – Model description
  – Inference
  – Handling overfitting

• Results
Monolingual Model

- A quick way to learn tokenization
- Easiest solution: substring counts
  - \( P(w_i) = \frac{\text{count}(w_i)}{\sum_k \text{count}(w_k)} \)
  - Single pass through corpus
  - Learning probability with EM overfits
  - Simple substring counts overfits as well
Overfitting

- Problem: Longer substrings are preferred under this model
- However, shorter tokens are more frequent in the real world
Solution to Overfitting

- Control token size with length factor

- $P(w_i) \propto count(w_i) \phi(|w_i|)$

- Geometric distribution would be a natural choice:
  
  - $\phi_1(\ell) = P(s)(1 - P(s))^{\ell-1}$

- Observation: Heavier penalty for longer tokens is desired

- Doubly exponential length factor:
  
  - $\phi_2(\ell) = 2^{-\ell\lambda}$
Length Factor

- Length factor vs. empirical token length distribution

![Graph showing length factor vs. empirical token length distribution with a peak at P(s) = 0.58 and lambda = 2.13 for Korean tokens.](image-url)
Setting the Parameter

- Both $\phi_1$ and $\phi_2$ have a single parameter
- We set the parameter such that number of tokens in the half of the parallel corpus match the other half
- Justification:
  - Hypothesis: Tokenizing this way will produce tokens that are closer to an ideal situation thus result in better MT system
  - Ideal case: one-to-one correspondence between tokens of two languages
Related Work

• Goldwater et al. (2006) use geometric distribution as base distribution for Dirichlet process in their Bayesian segmentation model to model word acquisition in infants

• Liang and Klein (2009) use doubly exponential length factor in their word segmentation model to test an online EM algorithm

• Chang et al. (2008) use a feature in their CRF Chinese segmenter to tweak average size of tokens to improve MT performance
Bilingual Model

• Can we learn segmentation of one language from the other language in parallel corpus?

• Our generative Model:

  \[ c_1 \quad c_2 \quad c_3 \]

  \[ c_4 \]

  \[ f_1 \quad f_2 \]

  \[ e_1 \quad e_2 \]

  The model learns alignments and segmentation is by-product of alignments
Inference

• The model uses IBM word alignment model 1
  
  \[ P(f|e) = \prod_i \sum_j P(f_i | e_j) P(a_i = j) \]

• \( f \) is unknown
  
  \[ f = s \circ c \]

• Apply dynamic programming over hidden segmentation \( s \)
  
  Analogous to HMM’s forward-backward algorithm
  
  Transition: segmentation
  
  Emission: alignment
Forward-Backward Algorithm

\[ \alpha(i) = P(c_1^i, s_i = 1 \mid e) \]

\[ \beta(j) = P(c_j^m, s_j = 1 \mid e) \]

\[ P(c_{i+1}^j, a = k \mid e) = \frac{\alpha(i) P(c_{i+1}^j \mid e_k) P(a = k) \beta(j)}{P(c \mid e)} \]
Overfitting

- We know the solution has to be **very** sparse
  - Solution: use a sparse prior
    \[
    \theta_e \mid \alpha \sim \text{Dir}(\alpha),
    \]
    \[
    f_i \mid e_i = e \sim \text{Multi}(\theta_e).
    \]
  - Use VB: minor change to inference (Johnson 2007)

- Further controlling overfitting with length factor
  - \( \phi_1 \) can be embedded in the model
    (the parameter can be learned)
  - \( \phi_2 \) can be used in the same manner as the first model
Related Work

• Kikui and Yamamoto (2002) use similar word alignment-based unsupervised segmentation to find new translation pairs from untokenized corpus

• Xu et al. (2008) use similar word alignment-based segmentation model (using Gibbs sampling for inference) as part of their Chinese word segmenter
Summary of Models

• Both models are unigram segmentation model
• Both models have explicit means to control size of tokens
• Monolingual model uses substring count to estimate $P(f)$
• Bilingual model uses word alignment to estimate
  \[ P(f) = \sum_e P(f \mid e)P(e) \]
• Both models use the Viterbi algorithm to find the best segmentation according to $P(f)$
• Both models limit maximum size of $f$ for practical reasons
Experiments

• MT Systems for Chi-Eng, and Kor-Eng language pairs
  – 2M words on English side for both language pairs
  – monolingual/bilingual models with length factors
  – Moses (Koehn et al., 2007)

• Three Questions:
  – How do the models compare to other tokenization?
  – What are the effects of length factors?
  – Does bilingual model learn to segment better?
## Comparison to Supervised Segmentation

<table>
<thead>
<tr>
<th>Supervised</th>
<th>Chinese</th>
<th>Korean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule-based morphological analyzer</td>
<td></td>
<td>7.27</td>
</tr>
<tr>
<td>LDC segmenter</td>
<td>20.03</td>
<td></td>
</tr>
<tr>
<td>Xue’s segmenter</td>
<td>23.02</td>
<td></td>
</tr>
<tr>
<td>Stanford segmenter (pku)</td>
<td>21.69</td>
<td></td>
</tr>
<tr>
<td>Stanford segmenter (ctb)</td>
<td>22.45</td>
<td></td>
</tr>
<tr>
<td><strong>Unsupervised</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bilingual model with $\phi_1$ $P(s) = 0.9$</td>
<td>20.75</td>
<td>7.46</td>
</tr>
<tr>
<td>Bilingual model with $\phi_2$</td>
<td>22.31</td>
<td>7.35</td>
</tr>
</tbody>
</table>
Effect of length factor

Chinese
## Monolingual vs. Bilingual

<table>
<thead>
<tr>
<th>Model Description</th>
<th>Chinese</th>
<th>Korean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bilingual model with $\phi_1$ $P(s) = learned$</td>
<td>20.04</td>
<td>7.06</td>
</tr>
<tr>
<td>Bilingual model with $\phi_1$ $P(s) = 0.9$</td>
<td>20.75</td>
<td>7.46</td>
</tr>
<tr>
<td>Bilingual model with $\phi_1$ $P(s) = 0.7$</td>
<td>20.59</td>
<td>7.31</td>
</tr>
<tr>
<td>Bilingual model with $\phi_1$ $P(s) = 0.5$</td>
<td>19.68</td>
<td>7.18</td>
</tr>
<tr>
<td>Bilingual model with $\phi_1$ $P(s) = 0.3$</td>
<td>20.02</td>
<td>7.38</td>
</tr>
<tr>
<td>Bilingual model with $\phi_2$</td>
<td>22.31</td>
<td>7.35</td>
</tr>
<tr>
<td>Monolingual model with $\phi_1$</td>
<td>20.93</td>
<td>6.76</td>
</tr>
<tr>
<td>Monolingual model with $\phi_2$</td>
<td>20.72</td>
<td>7.02</td>
</tr>
</tbody>
</table>
Summary

• Unsupervised tokenization methods are comparable to supervised ones in use for MT

• Bilingual model does learn better token probability for MT

• Heavier penalty for longer tokens is a useful means to prevent overfitting in segmentation

• Need to optimize parameters for end-to-end translation quality
## Additional Results

<table>
<thead>
<tr>
<th>Monolingual model with $\phi_1$ (EM)</th>
<th>15.70</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monolingual model with $\phi_2$ (EM)</td>
<td>21.30</td>
</tr>
<tr>
<td>Monolingual model with $\phi_1$</td>
<td>20.93</td>
</tr>
<tr>
<td>Monolingual model with $\phi_2$</td>
<td>20.72</td>
</tr>
</tbody>
</table>