An Overview of Transliteration

Dezhong Deng
the Graduate Center, CUNY
dengdezhong816@gmail.com

Wei Duo Li Ya
维多利亚

VICTORIA

human language technology
center of excellence

CUNY
The City University of New York
Transliteration is Not Trivial

• how many ways to transliterate?

[hIr]

Jie Ke
杰克
Jack
Jaque
Jake
Transliteration is Not Trivial

- when to transliterate?

![Library of Alexandria](image1)

- (No)
- (Yes)

![Seven Years Later](image2)

史蒂芬.耶尔斯.莱特
(person name)

![Tibet != Taipei](image3)

Tibet != Taipei
Can we design algorithms for transliteration that use context information like human?

- Our idea: train features from data to capture context
- Challenge 1: how to capture various of information (local vs. global)?
  - a model that can handle global features is required
- Challenge 2: how to learn features from big data?
  - faster training is required: $O(n)$ on number of training examples
Human vs. Computer Trans

- Can we learn the hidden structure behind transliteration?
  - Our idea: we learn alignments from data!
  - Challenge 1: how to train without knowing alignments?
    - we need to automatically get alignments over training
  - Challenge 2: how to deal with ambiguity explosion?
    - faster decoding on a single example is required
In This Talk..

- We review generative transliteration approaches
  - alignment: from single to multiple
    - allow various types of alignments
    - suits most of the languages
  - training: on to the linear
    - Faster training; suits on big data
  - identifier: transliterate or not
    - especially for grapheme-grapheme transliteration
What is Transliteration

- Is a Conversion across pairs with different alphabets
- Focus on phonetic translation
  - phoenix $\rightarrow$ [finlks] (Grapheme - Grapheme)
  - 希拉里 克林顿 $\rightarrow$ Hillary Clinton (Grapheme - Phoneme)
- Transliteration as a classification problem
  - Focus on learning mechanism to train a model that generates transliteration outputs
- Transliteration as a sequence modeling problem
  - Focus on discover hidden structures between transliterating from source to target
Transliteration Alignments

- the hidden structure behind transliteration
- Many-to-Many alignment is allowed in almost all kinds of transliteration

<table>
<thead>
<tr>
<th>rule</th>
<th>example</th>
<th>word</th>
</tr>
</thead>
<tbody>
<tr>
<td>s → t</td>
<td>u → AH</td>
<td>duck</td>
</tr>
<tr>
<td>s → ∅</td>
<td>e → ∅</td>
<td>home</td>
</tr>
<tr>
<td>∅ → t</td>
<td></td>
<td></td>
</tr>
<tr>
<td>s → t t</td>
<td>x → K S</td>
<td>max</td>
</tr>
<tr>
<td>s s → t</td>
<td>ch → CH</td>
<td>beach</td>
</tr>
</tbody>
</table>

- Alignment disambiguation
  - Combining Inner structure (source) and Outer structure (context & previous alignments)
  - Adding global features
Weighted Finite State Automata

- proposed by Kevin Knight in 1998
- generally one-to-one alignment
- use all training pairs to create a huge FSM
EM-Based Approaches

- Daelemans 97; Black 98; Jiampojamarn 07

**Algorithm 1** Generalized EM algorithm for alignment training

Input: transliteration pair training set \( S \)
Output: alignment probability table \( P(s, t) \) with source character substring \( s \) and target character substring \( t \)

1: Initialize probability table \( P(s, t) \)
2: repeat
3: for each pair \( (x, y) \) in \( S \) do
4: for each alignment path \( a = [(s_1, t_1), (s_2, t_2), ..., (s_k, t_k)] \) in \( (x, y) \) do
5: calculate weight \( W(a) \) using \( P(s, t) \) on alignments of \( a \)
6: end for
7: end for
8: Recalculate \( P(s, t) \) using count-and-divide on \( W(a) \) for all alignment paths
9: until convergence on \( P(s, t) \)

- alignment: many-to-many; slow training time
Structured Perceptron

- online-learning: one example at a time
- learning by doing
  - find the best output under the current weights
  - update weights at mistakes
Structured Perceptron

- binary classification
  - constant # of classes
  - exact inference
  - update weights if $y \neq z$

- structured classification
  - exponential # of classes
  - hard inference
  - update weights if $y \neq z$

- challenge: search efficiency (exponentially many classes)
  - often use dynamic programming (DP)
  - but still too slow for repeated use, e.g. parsing is $O(n^3)$
  - and can’t use non-local features in DP
Perceptron \(w/ \) Inexact Inference

**BILINGUAL**

**BAYLINGNG GW AH L**

\( x \rightarrow \text{inexact inference} \rightarrow z \rightarrow \text{update weights if } y \neq z \)

- routine use of inexact inference in NLP (e.g. beam search)
- how does structured perceptron work with inexact search?
  - standard perceptron: search errors break learning properties
  - Collins 04: found early perceptron can converge (a partial answer)
  - Huang 12: propose the max-violation perceptron, and prove the convergence of all the violation-fixing perceptrons (including early perceptron) (the final answer)

**Greedy search**

**Beam search**

does it still work???

**standard perceptron NO**

**violation-fixing perceptron YES**
Max-Violation Perceptron

- early perceptron learns too slow due to partial updates
- max-violation: use the prefix where violation is maximum
  - “worst-mistake” in search space
  - converges faster than the others
Learning with Latent Variables

- aka “weakly-supervised” or “partially-observed” learning
- learning from “natural annotations”; more scalable
- examples: translation, transliteration, semantic parsing...

parallel text

Bush talked with Sharon

parallel text

(Dezhong Deng, 2006; Yu et al, 2013; Xiao and Xiong, 2013)

transliteration

コンピューター

computer

transliteration

(Dezhong Deng, 2006; Yu et al, 2013; Xiao and Xiong, 2013)

QA pairs

What is the largest state?

argmax(state, size)

Alaska

QA pairs

(Dezhong Deng, 2006; Yu et al, 2013; Xiao and Xiong, 2013)
Force-Decoding in Perceptron

- Latent structured perceptron
- no explicit positive signal
  - hallucinate the “correct” derivation by current weights
- update using highest scoring gold derivation from highest scoring derivation

\[ d^* \]

training example

\[
\begin{array}{c}
\text{BILINGUAL} \\
\text{forced decoding space}
\end{array}
\]

\[
\begin{array}{c}
\text{BAYLINGWHAH} \\
\implies
\text{y ≠ z}
\end{array}
\]

during online learning...

\[
\begin{array}{c}
\text{BILINGUAL} \\
\text{full search space}
\end{array}
\]

\[
\begin{array}{c}
\text{BYLINGGYUW} \\
\implies
\text{highest-scoring derivation}
\end{array}
\]

(Liang et al 2006; Yu et al 2013)
Entity Transliteration: G2G vs. G2P

- A grapheme-grapheme transliteration task
- but can be treated as grapheme-phoneme transliteration
  - Hillary Clinton → 希拉里 克林顿
    [hiˈləri ˈklinkən] → Xi La Li Ke Lin Dun
- grapheme to phoneme in a single language
  - Chinese: easy, almost deterministic
  - English: hard, with lots of ambiguity
    - hard to rely on a G2P transliteration model because of the high error rate (Jiampojamarn 11: ~76% accuracy)
- from Chinese to English: phoneme to grapheme
- from English to Chinese: grapheme to phoneme/grapheme
Multi-Label Transliteration

- Entity transliteration could be a multi-label transliteration task
  - 杰克 → Jack/Jaque/Jake..
  - Jaina → 吉安娜/珍娜..

- Reason: words from different language sources, especially named entities

- Training with one-to-many pairs
  - choose the nearest golden target from the decoding target by calculating edit distance
Thanks for listening!

Dezhong Deng
the Graduate Center, CUNY
dengdezhong816@gmail.com