Multilingual Learning

Anoop Kunchukuttan

Microsoft AI and Research

Last updated 20th September 2018
Broad Goal: Build NLP Applications that can work on different languages
Facets of an NLP Application

- Algorithms
- Knowledge
- Data
Facets of an NLP Application

**RULE-BASED SYSTEMS**

- Algorithms
- Knowledge
- Data

**Algorithms**
- Expert Systems
- Theorem Provers
- Parsers
- Finite State Transducers

**Knowledge**
- Rules for morphological analyzers, Production rules, etc.
- Lot of linguistic knowledge encoded

**Data**
- Paradigm Tables, dictionaries, etc.
- Lot of linguistic knowledge encoded

Some degree of language independence through good software engineering and knowledge of linguistic regularities
Facets of an NLP Application

**STATISTICAL ML SYSTEMS (Pre-Deep Learning)**

- **Algorithms**
  - Largely language independent, could solve non-trivial problems efficiently
  - Supervised Classifiers
  - Sequence Learning Algorithms
  - Probabilistic Parsers
  - Weighted Finite State Transducers

- **Knowledge**
  - Lot of linguistic knowledge encoded
  - Feature engineering is easier than maintain rules and knowledge-bases

- **Data**
  - Annotated Data, Paradigm Tables, dictionaries, etc.
  - Lot of linguistic knowledge encoded

**Feature Engineering**
- General language-independent ML algorithms and easy feature learning
Facets of an NLP Application

**DEEP LEARNING SYSTEMS**

- Algorithms
  - Largely language independent
  - Fully Connected Networks
  - Recurrent Networks
  - Convolutional Neural Networks
  - Sequence-to-Sequence Learning

- Knowledge
  - Representation Learning, Architecture Engineering, AutoML
  - Very little knowledge; annotated data is still required

- Data
  - Annotated Data, Paradigm Tables, dictionaries, etc.
  - Very little knowledge; annotated data is still required

Feature engineering is unsupervised, largely language independent

Neural Networks provide a convenient language for expressing problems, representation learning automated feature engineering
Facets of an NLP Application

**DEEP LEARNING SYSTEMS**

- **Algorithms**
  - Largely language independent
  - Fully Connected Networks
  - Recurrent Networks
  - Convolutional Neural Networks
  - Sequence-to-Sequence Learning

- **Knowledge**
  - Representation Learning, Architecture Engineering, AutoML
  - Very little knowledge; annotated data is still required

- **Data**
  - Annotated Data, Paradigm Tables, dictionaries, etc.

Feature engineering is unsupervised, largely language independent.

Neural Networks provide a convenient language for expressing problems, representation learning automated feature engineering.
Focus of today’s session

How to leverage data for one language to build NLP applications for another language?
Multilingual Learning Scenarios

**Joint Learning**

- Analogy to Multi-task learning \( \rightarrow \) Task \( \equiv \) Language
- Related Tasks can share representations
- **Representation Bias**: Learn the task to generalize over multiple languages
- Eavesdropping
- **Data Augmentation**

(Caruana., 1997)
Multilingual Learning Scenarios

**Transfer Learning**

Low resource language can benefit from data for high resource language

(Caruana., 1997)
Multilingual Learning Scenarios

**Zero-shot Learning**

Can system be trained for one language so that they work out of the box for another language?
What does Deep Learning bring to the table?

• Neural Networks provide a powerful framework for Multilingual learning
  • Caruana’s seminal work on Multi-task learning in 1997 used Neural Networks

• Word embeddings: Powerful feature representation mechanism to capture syntactic and semantic similarities
  • Distributed representation
  • Unsupervised learning

• Algebraic reasoning as opposed to Mathematical Logic

• Numerical optimization as opposed to combinatorial optimization
A Typical Multilingual NLP Pipeline

Text → Tokens → Token Embeddings → Application specific Deep Neural Network layers → Output (text or otherwise) → Text Embedding
A Typical Multilingual NLP Pipeline

- **Text**
- **Tokens**
- **Token Embeddings**
- **Text Embedding**
- **Output** (text or otherwise)
- **Application specific Deep Neural Network layers**

Similar tokens across languages should have similar embeddings.
A Typical Multilingual NLP Pipeline

Text

Tokens

Token Embeddings

Output
(text or otherwise)

Application specific Deep Neural Network layers

Text Embedding

Similar text across languages should have similar embeddings
A Typical Multilingual NLP Pipeline

Pre-process to facilitate similar embeddings across languages?

Text

Tokens

Token Embeddings

Output (text or otherwise)

Application specific Deep Neural Network layers

Text Embedding
A Typical Multilingual NLP Pipeline

Text → Tokens → Token Embeddings → Application specific Deep Neural Network layers → Output (text or otherwise)

How to support multiple target languages?
Outline

• Learning Cross-lingual Embeddings

• Training a Multilingual NLP Application

• Related Languages and Multilingual Learning

• Summary and Research Directions
Cross-Lingual Embeddings

Offline Methods
Online Methods
Some observations
Evaluation
Unsupervised Learning
Monolingual Word Representations
(capture syntactic and semantic similarities between words)

$\text{embed}(y) = f(\text{embed}(x))$

$x, y$ are source and target words
$\text{embed}(w)$: embedding for word $w$

Multilingual Word Representations
(capture syntactic and semantic similarities between words both within and across languages)

(Source: Khapra and Chandar, 2016)
Is it possible to learn mapping functions?

• Languages share concepts grounded in the real world
• Some evidence of universal semantic structure (Youn et al., 2016)
• Isomorphism between embedding spaces (Mikolov et al., 2013)
• Isomorphism can be captured via a linear transformation

(Source: Mikolov et al., 2013)
Offline Methods

Learn monolingual and cross-lingual embeddings **separately**

General require weaker parallel signals

*e.g., bilingual dictionaries*

Online Methods

Learn monolingual and cross-lingual embeddings **jointly**

Generally require stronger parallel signals

*e.g., parallel corpus*
Cross-Lingual Embeddings

- Offline Methods
- Online Methods
- Some observations
- Evaluation
- Unsupervised Learning
Supervised Learning

\[ XW = Y \]
Least Squares Solution \hfill (Mikolov et al., 2013)

\[ W^* = \text{argmin}_{W \in \mathbb{R}^d} \|XW - Y\|_2^2 \]

We can have a closed form solution:

\[ X^+ = (X^T X)^{-1} X^T \]

\[ W^* = X^+ Y \]

Solutions can be regularized using $L_1$ or $L_2$ norms to prevent overfitting
Orthogonality Constraint on $W$

$$W^T W = I$$

- Preserves similarity in the target space \citep{artetxe2016}: 
  
  $$(Wx)^T (Wy) = x^T W^T Wy = x^T y$$

- Mapping Function is reversible \citep{smith2017}:  
  
  $$W^T W x = x$$

- If source embeddings are unit vectors, orthogonality ensures target is also a unit vector \citep{xing2015}:  
  
  $$y^T y = (Wx)^T (Wx) = x^T W^T W x = x^T x = 1$$

- Why length normalize? $\Rightarrow$ dot product equivalent to cosine similarity
Orthogonal Procrustes Problem

\[
W^* = \arg\min_{W \in O^d} \|XW - Y\|_2^2
\]

We can have a closed form solution to this problem too (Schönemann, 1966)

\[
Y^T X = U\Sigma V^T
\]

\[
W^* = VU^T
\]

If embeddings are length-normalized, the above objective is equivalent to maximizing cosine similarity

\[
W^* = \arg\max_{W \in O^d} \sum_i \cos(X_i W, Y_i)
\]
Canonical Correlation Analysis (CCA)  

(Faruqui and Dyer, 2014; Ammar et al. 2015)

Regression methods ➔ maximize similarity between target & mapped source embeddings

An alternative way to compare:

Is there a latent space where the dimensions of the embeddings are correlated?
paanii → water
ghar → house
sadak → road
agni → fire
maximize \text{trace}((XA)^T(YB))

This term captures the correlation between the dimensions in the latent space defined by A and B.
Fine-tuning the bilingual mappings

**Strong assumptions**
- Linear Transformation
- Orthogonality constraint

Meeting in the middle
*(Doval et al. 2018)*

Learn a correction function $g: X' \rightarrow \text{avgY}$
Learning a distance metric in a latent space (GeoMM)

(Jawanpuria et al, 2018)

Latent Space

Scaling

Rotation

$B^{1/2}$ $B^{1/2}$

$W$ is factorized as

$W = VBU^T$

$B$ is learnt from data

Applies corrections in latent space

$X$ water

$Y$ water

पानी
Multilingual Embeddings

Represent embeddings from multiple languages in a single vector space

Map to a common pivot language
(Ammar et al, 2016)

Map to a latent space
(Jawanpuria et al, 2018; Yova et al, 2018)
Bilingual Lexicon Induction aka Word Translation

Given a mapping function and source/target words and embeddings:
Can we extract a bilingual dictionary?

\[ y' = W(\text{embed}(\text{paanii})) \]

\[ \max_{y \in Y} \cos(\text{embed}(y), y') \Rightarrow \text{water} \]

A standard intrinsic evaluation task for judging quality of cross-lingual embedding quality
The Hubness Problem with Nearest Neighbour

In high dimensional spaces, some points are neighbours of many points ➔ hubs

Adversely impacts Nearest Neighbour search ➔ especially in mapped spaces

Why does hubness occur?

• Points are closer in mapped space with least-squares?

• Pairwise similarities tend to converge to constant as dimensionality increases
Solutions to Hubness

Modify the search algorithm

- Inverted Rank (IR)
- Inverted Softmax (ISF)
- Cross-domain Similarity Local Scaling (CSLS)

Modify the learning objective to address hubness

- Max Margin Training
- Optimizing CSLS
Inverted Rank \hfill (Dinu et al., 2015)

\[ \text{Rank}_{a,Z}(z) : \text{Rank of } z \text{ in neighbourhood of } a \text{ w.r.t candidate nodes } Z \]

In nearest neighbor we pick the target of rank 1

\[ \text{NN}(x) = \arg\min_{y \in Y} \text{Rank}_{x,Y}(y) \]

In nearest neighbor we pick the target for which \( x \) has the lowest rank

\[ \text{IR}(x) = \arg\min_{y \in Y} \text{Rank}_{y,X}(x) \]

Kind of collective classification, hubs will be assigned to the \( x \) to which they are closest
Inverted Softmax  
*(Smith et al., 2017)*

Another way of inverse information lookup like IR

\[ P(y|x) = \frac{e^{\beta \cos(x,y)}}{\sum_{y'} e^{\beta \cos(x,y')}} \]  

**NN**

\[ P(y|x) = \frac{e^{\beta \cos(x,y)}}{\sum_{y'} e^{\beta \cos(x,y')}} \]

Distance Metric is generally normalized over target

\[ P(y|x) = \frac{e^{\beta \cos(x,y)}}{\alpha_y \sum_{y'} e^{\beta \cos(x',y)}} \]

**ISF**

Modified Distance Metric normalized over source

Will penalize hubs since they have a large denominator

Local scaling of the distance metric
Cross-domain Similarity Local Scaling (CSLS)

(Conneau et al., 2018)

Another Local scaling of the distance metric

Define mean similarity of a mapped source word to its target neighbourhood and vice versa

\[
\begin{align*}
r_T(x) &= \frac{1}{K} \sum_{y \in N_T(x)} \cos(x, y) \\
r_S(y) &= \frac{1}{K} \sum_{x \in N_S(y)} \cos(x, y)
\end{align*}
\]

\[
\text{CSLS}(x, y) = 2 \cos(x, y) - r_T(x) - r_S(y)
\]

Will penalize hubs since they have large mean similarity

Symmetric metric
No parameter tuning
Optimizing CSLS  

(Joulin et al., 2018)

For CSLS retrieval,

**Training Metric:** Cosine similarity  
**Test Metric:** CSLS

Mismatch between train and test metric

A good principle is to optimize for the objective we are interested in  ➔ optimize CSLS loss directly

\[
CSLS_{\text{loss}}(x, y) = -2 \cos(x, y) + r_T(x) + r_S(y)
\]
Max-Margin Formulation \textbf{(Lazaridou et al., 2015)}

\[
\sum_{j \neq i}^{N} \max \left\{ 0, \gamma + \|Wx_i - y_i\|^2 - \|Wx_i - y_j\|^2 \right\}
\]

Negative example must be as far good example as possible

Why would max-margin reduce hubness? \textbf{⇒} No clear answer
Cross-Lingual Embeddings

Offline Methods

Online Methods (Slides adapted from Khapra and Chandar, 2016)

Some observations

Evaluation

Unsupervised Learning
Using Parallel Corpus Only

Training data: Parallel sentences

\[ a = \text{English sentence} \]
\[ b = \text{parallel French sentence} \]
\[ n = \text{random French sentence} \]

\[ E(a, b) = \|f(a) - g(b)\|^2 \]

minimize
\[ \max(0, m + E(a, b) - E(a, n)) \]

Backpropagate & update \( w_i \)'s in both languages

To reduce the distance between \( f(a) \) & \( g(b) \) the model will eventually learn to reduce the distance between (chair, chaise), (sit, assis), (he, il) etc.
Using Parallel Corpus and Monolingual Corpus (Gouws et al., 2015)

Fr positive: Il était assis sur une chaise
Fr negative: Il était assis sur une oxygène

Independently update $\theta^e$ and $\theta^f$

$$\text{maximize max}(0, 1 - s^f + s^e)$$

w.r.t $\theta^e$

+ Parallel data

En: he sat on a chair $[s_e = w_1^e, w_2^e, w_3^e, w_4^e, w_5^e]$

Fr : Il était assis sur une chaise $[s_f = w_1^f, w_2^f, w_3^f, w_4^f, w_5^f]$

now, also minimize $\Omega(W_{emb}^e, W_{emb}^f) = \left\| \frac{1}{m} \sum_{w_i \in S^e} W_{emb_i}^e - \frac{1}{n} \sum_{w_j \in S^e} W_{emb_i}^f \right\|^2$

w.r.t $W_{emb}^e, W_{emb}^f$

$$\text{maximize max}(0, 1 - s^e + s^e)$$

w.r.t $\theta^f$

(Gouws et al., 2015)
- Autoencoder approach
- Correlation term is important to ensure common representation
- Combines:
  - word similarity (recall Procrustes!)
  - dimension correlation (recall CCA!)
A general framework for cross-lingual embeddings

\[
\text{maximize} \quad \sum_{j \in \{e,f\}} \sum_{i=1}^{T_j} -\log(P(w_i|w_{i-k}, \ldots, w_{i-1})) + \lambda \cdot \Omega(W_{emb}^e, W_{emb}^f)
\]

w.r.t. \( \theta_e, \theta_f \)

\( \theta_e = W_{emb}^e, W_h^e, W_{out}^e \)

\( \theta_f = W_{emb}^f, W_h^f, W_{out}^f \)

\( \Omega(W_{emb}^e, W_{emb}^f) = \sum_{w_i \in V^e} \sum_{w_j \in V^f} \text{sim}(w_i, w_j) \ast \text{distance}(W_{emb_i}^e, W_{emb_j}^f) \)

This weighted sum will be low only when similar words across languages are embedded close to each other

Offline embeddings also follow this framework, but they optimize the monolingual and bilingual objectives sequentially
Cross-Lingual Embeddings

Offline Methods
Online Methods
Some observations
Evaluation
Unsupervised Learning
Intrinsic Evaluation

• Bilingual Lexicon Induction

• Cross-language word similarity task

Mostly offline methods
# Bilingual Lexicon Induction

<table>
<thead>
<tr>
<th>Method</th>
<th>P@1</th>
<th>P@5</th>
<th>P@10</th>
<th>P@1</th>
<th>P@5</th>
<th>P@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ordinary Least Squares</td>
<td>33.8</td>
<td>48.3</td>
<td>53.9</td>
<td>24.9</td>
<td>41.0</td>
<td>47.4</td>
</tr>
<tr>
<td>OP + NN</td>
<td>36.9</td>
<td>52.7</td>
<td>57.9</td>
<td>32.2</td>
<td>49.6</td>
<td>55.7</td>
</tr>
<tr>
<td>OP + IR</td>
<td>38.5</td>
<td>56.4</td>
<td>63.9</td>
<td>24.6</td>
<td>45.4</td>
<td>54.1</td>
</tr>
<tr>
<td>OP + ISF</td>
<td>43.1</td>
<td>60.7</td>
<td>66.4</td>
<td>38.0</td>
<td>58.5</td>
<td>63.6</td>
</tr>
<tr>
<td>OP + CSLS</td>
<td>44.9</td>
<td>61.8</td>
<td>66.6</td>
<td>38.5</td>
<td>57.2</td>
<td>63.0</td>
</tr>
<tr>
<td>OP + CSLS (optimize)</td>
<td><strong>45.3</strong></td>
<td>NA</td>
<td>NA</td>
<td>37.9</td>
<td>NA</td>
<td>NA</td>
</tr>
<tr>
<td>CCA</td>
<td>36.1</td>
<td>52.7</td>
<td>58.1</td>
<td>31.0</td>
<td>49.9</td>
<td>57.0</td>
</tr>
</tbody>
</table>

*Orthogonality constraint helps*
## Bilingual Lexicon Induction

<table>
<thead>
<tr>
<th></th>
<th>English to Italian</th>
<th>Italian to English</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P@1</td>
<td>P@5</td>
</tr>
<tr>
<td>Ordinary Least Squares</td>
<td>33.8</td>
<td>48.3</td>
</tr>
<tr>
<td>OP + NN</td>
<td>36.9</td>
<td>52.7</td>
</tr>
<tr>
<td>OP + IR</td>
<td>38.5</td>
<td>56.4</td>
</tr>
<tr>
<td>OP + ISF</td>
<td>43.1</td>
<td>60.7</td>
</tr>
<tr>
<td>OP + CSLS</td>
<td>44.9</td>
<td>61.8</td>
</tr>
<tr>
<td>OP + CSLS (optimize)</td>
<td>45.3</td>
<td>NA</td>
</tr>
<tr>
<td>CCA</td>
<td>36.1</td>
<td>52.7</td>
</tr>
</tbody>
</table>

- Modified retrieval significantly improve performance over vanilla Nearest Neighbour Search
- CSLS is best performing
- Optimizing CSLS loss also gives some improvements
### Bilingual Lexicon Induction

<table>
<thead>
<tr>
<th></th>
<th>English to Italian</th>
<th>Italian to English</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P@1</td>
<td>P@5</td>
</tr>
<tr>
<td>Ordinary Least Squares</td>
<td>33.8</td>
<td>48.3</td>
</tr>
<tr>
<td>OP + NN</td>
<td>36.9</td>
<td>52.7</td>
</tr>
<tr>
<td>OP + IR</td>
<td>38.5</td>
<td>56.4</td>
</tr>
<tr>
<td>OP + ISF</td>
<td>43.1</td>
<td>60.7</td>
</tr>
<tr>
<td>OP + CSLS</td>
<td>44.9</td>
<td>61.8</td>
</tr>
<tr>
<td>OP + CSLS (optimize)</td>
<td><strong>45.3</strong></td>
<td>NA</td>
</tr>
<tr>
<td>CCA</td>
<td>36.1</td>
<td>52.7</td>
</tr>
</tbody>
</table>

*Orthogonal Procrustes solution and CCA give roughly the same results*
Extrinsic Evaluation

• Cross-lingual Document Classification

• Cross-lingual Dependency Parsing

*Mostly online methods*
### Cross-lingual Document Classification

<table>
<thead>
<tr>
<th>Approach</th>
<th>en → de</th>
<th>de → en</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hermann &amp; Blunson, 2014</td>
<td>83.7</td>
<td>71.4</td>
</tr>
<tr>
<td>Chandar et al., 2014</td>
<td>91.8</td>
<td>72.8</td>
</tr>
<tr>
<td>Gouws et al., 2015</td>
<td>86.5</td>
<td>75.0</td>
</tr>
</tbody>
</table>

Leveraging monolingual and parallel corpora yields better results.
Cross-Lingual Embeddings

Offline Methods
Online Methods
Some observations
Evaluation
Unsupervised Learning
More observations on different aspects of the problem

Take them with a pinch of salt, since comprehensive experimentation is lacking

More like rule of thumb to make decisions
Effect of bilingual dictionary size

<table>
<thead>
<tr>
<th>Dictionary Size</th>
<th>Precision@1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1K</td>
<td>20.09</td>
</tr>
<tr>
<td>5K</td>
<td>37.3</td>
</tr>
<tr>
<td>10K</td>
<td>37.5</td>
</tr>
<tr>
<td>20K</td>
<td>37.9</td>
</tr>
</tbody>
</table>

Beyond a certain size, the size of bilingual dictionary does not seem useful.

What if the bilingual dictionaries are really large?
Effect of monolingual corpora size

(Mikolov et al., 2013)

Large monolingual corpora substantially increases the quality of embeddings

Having large monolingual corpora may be more useful than having large bilingual dictionary?
How difficult is to translate less frequent words?

- Performance does not drop very sharply for intermediate frequency words
- Performance drops sharply for very rare words

(Mikolov et al., 2013)

(Dinu et al., 2015)

Note: GC is same as Inverse Rank retrieval
Do these approaches work for all languages?

https://github.com/Babylonpartners/fastText_multilingual#right-now-prove-that-this-procedure-actually-worked

• Study on 78 languages
• Trained on 10k words (Dictionary created using Google Translate)
• Tested on 2500 words
• Method described by Smith et al., 2017 (Procrustes with inverted softmax)

<table>
<thead>
<tr>
<th>Best Languages</th>
<th>Worst Languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>French</td>
<td>Urdu</td>
</tr>
<tr>
<td>Portuguese</td>
<td>Marathi</td>
</tr>
<tr>
<td>Spanish</td>
<td>Japanese</td>
</tr>
<tr>
<td>Norwegian</td>
<td>Punjabi</td>
</tr>
<tr>
<td>Dutch</td>
<td>Burmese</td>
</tr>
<tr>
<td>Czech</td>
<td>Luxembourghish</td>
</tr>
<tr>
<td>Hungarian</td>
<td>Malagasy</td>
</tr>
</tbody>
</table>

No patterns, seems to be a function of dictionary quality in each language
Facebook has recently provided high quality bilingual dictionaries ➔ a testbed to do better testing

https://github.com/facebookresearch/MUSE#ground-truth-bilingual-dictionaries
Do these approaches work for all languages?

Results on more languages from Conneau et al., 2018

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Procrustes - NN</td>
<td>77.4</td>
<td>77.3</td>
<td>74.9</td>
<td>76.1</td>
<td>68.4</td>
<td>67.7</td>
<td>47.0</td>
<td>58.2</td>
<td>40.6</td>
<td>30.2</td>
<td>22.1</td>
<td>20.4</td>
</tr>
<tr>
<td>Procrustes - ISF</td>
<td>81.1</td>
<td>82.6</td>
<td>81.1</td>
<td>81.3</td>
<td>71.1</td>
<td>71.5</td>
<td>49.5</td>
<td>63.8</td>
<td>35.7</td>
<td>37.5</td>
<td>29.0</td>
<td>27.9</td>
</tr>
<tr>
<td>Procrustes - CSLS</td>
<td>81.4</td>
<td>82.9</td>
<td>81.1</td>
<td>82.4</td>
<td>73.5</td>
<td>72.4</td>
<td>51.7</td>
<td>63.7</td>
<td>42.7</td>
<td>36.7</td>
<td>29.3</td>
<td>25.3</td>
</tr>
</tbody>
</table>

Seems to work well on mainland European languages compared to Russian, Chinese and Esperanto
Cross-Lingual Embeddings

Offline Methods
Online Methods
Some observations
Evaluation
Unsupervised Learning
Unsupervised Learning

\[ XW = PY \]

\[ P = \begin{bmatrix} 1 & & & \vphantom{1} \\ & 1 & & \vphantom{1} \\ & & 1 & \vphantom{1} \\ & & & 1 \end{bmatrix} \] (Permutation matrix)
Many language pairs may not have an available bilingual dictionary

**Mostly offline methods – by definition**

*Exciting developments on this task this year*
Starting with a small seed dictionary

(Artetxe et al., 2017)

• Semi-supervised solution
• As small as 50-100
• Dictionary can just be aligned digits and numbers
  • \( \text{१} \rightarrow 1 \)
  • \( \text{२८९} \rightarrow 289 \)
  • \( \text{५} \rightarrow 5 \)
• Identical strings
  • Requires both languages to have similar scripts and share vocabulary
• Bootstrapping solution
Enhancements by Hoshen and Wolf (2018)
- do away with the need for seed dictionary by matching principal components for initialization
- consider a objective in other direction and circular objective too

\[ W^* = \arg \max_W \sum_i \max_j (X_{i*}W) \cdot Z_{j*} \]

\[ \text{s.t. } WW^T = W^TW = I \]

Enhancements by Artetxe et al., (2018b)
- do away with the need for seed dictionary by using word similarity distribution for initialization
Bootstrapping works well with small dictionaries.

Aligned numbers are sufficient to bootstrap.

<table>
<thead>
<tr>
<th>Source</th>
<th>English-Italian</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th>English-German</th>
<th></th>
<th></th>
<th></th>
<th>English-Finnish</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5,000</td>
<td>25</td>
<td>num.</td>
<td>5,000</td>
<td>25</td>
<td>num.</td>
<td>5,000</td>
<td>25</td>
<td>num.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mikelov et al. (2013a)</td>
<td>34.93</td>
<td>0.00</td>
<td>0.00</td>
<td>35.00</td>
<td>0.00</td>
<td>0.07</td>
<td>25.91</td>
<td>0.00</td>
<td>0.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Xing et al. (2015)</td>
<td>36.87</td>
<td>0.00</td>
<td>0.13</td>
<td>41.27</td>
<td>0.07</td>
<td>0.53</td>
<td>28.23</td>
<td>0.07</td>
<td>0.56</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zhang et al. (2016)</td>
<td>36.73</td>
<td>0.07</td>
<td>0.27</td>
<td>40.80</td>
<td>0.13</td>
<td>0.87</td>
<td>28.16</td>
<td>0.14</td>
<td>0.42</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Artetxe et al. (2016)</td>
<td>39.27</td>
<td>0.07</td>
<td>0.40</td>
<td><strong>41.87</strong></td>
<td>0.13</td>
<td>0.73</td>
<td><strong>30.62</strong></td>
<td>0.21</td>
<td>0.77</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Artetxe et al. (2017)</td>
<td><strong>39.67</strong></td>
<td><strong>37.27</strong></td>
<td><strong>39.40</strong></td>
<td><strong>40.87</strong></td>
<td><strong>39.60</strong></td>
<td><strong>40.27</strong></td>
<td><strong>28.72</strong></td>
<td><strong>28.16</strong></td>
<td><strong>26.47</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: Artetxe et al., (2017)
Adversarial Training

(Barone, 2016; Zhang et al., 2017a,b; Conneau et al., 2018)

We want to make $W_x$ and $y$ indistinguishable

Step 1: Make a good discriminator that can distinguish between $W_x$ and $y$ (optimize $\theta_D$)

Step 2: Try to fool this discriminator by generating $W_x$ which are indistinguishable (optimize $\theta_G$)

Iterate with improved generator

Conneau et al., 2018 suggested multiple runs, rebuilding & refining dictionary after each run
Tips for training

• Training adversarial networks is not easy – have to balance two objectives

• There may be a mismatch between discriminator and task classifier quality

• *e.g.* If the discriminator is weaker
  • Design training schedule s.t. early epochs focus on improving the classifier

• Stabilizing GAN training is an active area of work
Wasserstein Procrustes

\[ X \times W = PY \]

(Permutation matrix)

\[ P = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \]

(Zhang et al., 2017b; Grave et al., 2018)

\text{paanii} \quad \text{ghar} \quad \text{sadak} \quad \text{agni}

\text{road} \quad \text{house} \quad \text{water} \quad \text{fire}
If $P$ is known, we can find $W$ using the orthogonal Procrustes solution

$$W^* = \arg\min_{W \in O_d} \|XW - PY\|_2^2$$

If $W$ is known, finding $P$ is equivalent to finding maximum weight matching in a bipartite graph

The dataset as a whole is aligned, considering constraints from all examples.
Overall, problem is

$$\min_{W \in \mathcal{O}_d} \min_P \|XW - PY\|_2^2$$

We can solve each minimization problem alternately, keep the other parameter constant.

Good initialization of the problem is important.

Grave et al., 2018 suggest a convex relaxation of the above problem.

The solution to the convex relaxation is a good initializer to the problem.
Comparing unsupervised methods

- Unsupervised methods can rival supervised approaches
- Even linear transformation based methods can perform well
- Shows the strong structural correspondence between embedding spaces across languages
- A launchpad for unsupervised sentence translation

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Procrustes</td>
<td>82.7</td>
<td>84.2</td>
<td>82.7</td>
<td>83.4</td>
<td>74.8</td>
<td>73.2</td>
<td>51.3</td>
<td>63.7</td>
</tr>
<tr>
<td>Adversarial*</td>
<td>81.7</td>
<td>83.3</td>
<td>82.3</td>
<td>82.1</td>
<td>74.0</td>
<td>72.2</td>
<td>44.0</td>
<td>59.1</td>
</tr>
<tr>
<td>ICP*</td>
<td>82.1</td>
<td>84.1</td>
<td>82.3</td>
<td><strong>82.9</strong></td>
<td>74.7</td>
<td>73.0</td>
<td><strong>47.5</strong></td>
<td><strong>61.8</strong></td>
</tr>
<tr>
<td>Wasserstein Procrustes</td>
<td><strong>82.8</strong></td>
<td><strong>84.1</strong></td>
<td><strong>82.6</strong></td>
<td><strong>82.9</strong></td>
<td><strong>75.4</strong></td>
<td><strong>73.3</strong></td>
<td>43.7</td>
<td>59.1</td>
</tr>
</tbody>
</table>

Source: Grave et al., (2018)
Outline

• Learning Cross-lingual Embeddings

• *Training a Multilingual NLP Application*

• Related Languages and Multilingual Learning

• Summary and Research Directions
Embed - Encode - Attend - Decode Paradigm

(Bahdanau et al, 2015)
Joint Learning
Minimal Parameter Sharing

(Firat et al., 2016)

Separate vocabularies and embeddings
Embeddings learnt during training
Source Embeddings projected to a common space
Cycle through each language pair in minibatches
All Shared Architecture

(Johnson et al., 2017)

Shared vocabularies and embeddings across languages

Embeddings learnt during training

Source Embeddings projected to a common space

A minibatch contains data from all language pairs
How do we support multiple target languages with a single decoder?

A simple trick!

Append input with special token indicating the target language

For English-Hindi Translation

**Original Input:** France and Croatia will play the final on Sunday

**Modified Input:** France and Croatia will play the final on Sunday <hin>
Transfer Learning
Shared Encoder

(Zoph et al., 2016; Nguyen et al., 2017; Lee et al., 2017)
Shared Encoder

(Zoph et al., 2016; Nguyen and Chang, 2017; Lee et al., 2017)

Zoph et al., 2016: Randomly map primary and assisting language word embeddings

Lee et al., 2017: Character as basic unit
   Single vocabulary as long as primary and assisting languages have compatible scripts

Nguyen et al., 2017: Use BPE to learn a common vocabulary across primary and assisting languages
   BPE identifies small substring patterns in text
Use pretrained multilingual embeddings
How do we ensure that encoder representations are similar across languages?
Inexact mapping with bilingual embedding

(Xie et al., 2018)

Model may go astray due to embedding gap at the input

Replace word by its translation
Addressing word order divergence
Pre-ordering assisting language sentences

(Lot of work on source reordering like Ramanathan et al 2009, Ponti et al 2018; none for multi-linguality)
**Position independent encoder representations**

*(Xie et al., 2018)*

**Problem**: RNN architectures are sensitive to word-order

*Can we use an encoder representation that is not sensitive to the word order for the supporting language?*

*The Transformer architecture that uses Self-attention*
Shared Encoder with Adversarial Training

(Joty et al., 2017)

Hindi

Bengali

Telugu

E₁

E₂

E₃

Map Embeddings

Shared Encoder

Shared Attention Mechanism

Decoder

Language Discriminator

Generate embeddings which the language discriminator cannot distinguish

Keep improving the discriminator such that it is difficult to fool it

Joty et al., 2017
Training Process

Minibatch containing a mixture of Primary and Assisting language samples

- Freeze discriminator parameters
- Find TM model parameters that minimize $L_c(\theta)$ and maximize $L_l(\theta)$
- Find classifier parameters that minimize $L_l(\theta)$
- Freeze TM model parameters
Data Selection

(Rudramurthy et al., 2018)

Is all the high-resource assisting language data useful?
Maybe, sentences with a very different structure from primary language are harmful.
Let’s take a simpler example → Named Entity Recognition
Filter out training examples with high tag distribution divergence

<table>
<thead>
<tr>
<th>Word</th>
<th>Per</th>
<th>Loc</th>
<th>Org</th>
<th>Misc</th>
</tr>
</thead>
<tbody>
<tr>
<td>China</td>
<td>-</td>
<td>91</td>
<td>7</td>
<td>-</td>
</tr>
<tr>
<td>France</td>
<td>-</td>
<td>123</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Reuters</td>
<td>-</td>
<td>40</td>
<td>18</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Word</th>
<th>Per</th>
<th>Loc</th>
<th>Org</th>
<th>Misc</th>
</tr>
</thead>
<tbody>
<tr>
<td>China</td>
<td>-</td>
<td>20</td>
<td>49</td>
<td>1</td>
</tr>
<tr>
<td>France</td>
<td>-</td>
<td>-</td>
<td>10</td>
<td>-</td>
</tr>
<tr>
<td>Reuters</td>
<td>-</td>
<td>3</td>
<td>1</td>
<td>-</td>
</tr>
</tbody>
</table>
Training Transfer learning systems

Method 1

Sample from Parallel Corpora

Combine Parallel Corpora

Train

Method 2

Train

Model for C₂

Finetune

Model tuned for C₁
Zeroshot translation

*Can we translate language pairs we have not seen so far?*

- Unseen language pair
- Unseen source language
- Unseen target language
With a shared encoder, unseen source languages can be supported

Supporting unseen target languages is a challenge
Outline

• Learning Cross-lingual Embeddings

• Training a Multilingual NLP Application

• Related Languages and Multilingual Learning

• Summary and Research Directions
Related Languages (plus)
Pre-processing Text
Multi-task learning is more beneficial when tasks are related to each other
Related Languages

Related by Genealogy

Language Families
Dravidian, Indo-European, Turkic

Related by Contact

Linguistic Areas
Indian Subcontinent, Standard Average European

Related languages may not belong to the same language family!
<table>
<thead>
<tr>
<th>Lexical: share significant vocabulary (cognates &amp; loanwords)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Morphological: correspondence between suffixes/post-positions</th>
</tr>
</thead>
</table>

| Syntactic: share the same basic word order |
Why are we interested in such related languages?
These related languages are generally geographically contiguous
• 5 language families (+ 2 to 3 on the Andaman & Nicobar Islands)
• 22 scheduled languages
• **11 languages with more than 25 million speakers**
• Highly multilingual country
Naturally, a lot of communication between such languages (government, social, business needs)

Most translation requirements also involves related languages

**Between related languages**
- Hindi-Malayalam
- Marathi-Bengali
- Czech-Slovak

**Related languages** ↔ **Link languages**
- Kannada, Gujarati → English
- English → Tamil, Telugu

We want to be able to handle a large number of such languages
e.g. 30+ languages with a speaker population of 1 million + in the Indian subcontinent
Utilizing Lexical Similarity

Lexically Similar Languages
(Many words having similar **form** and **meaning**)

- **Cognates**
  - *a common etymological origin*
    
    | roTI (hi) | roTIA (pa) | bread |
    | bhai (hi) | bhAU (mr)  | brother |

- **Loan Words**
  - *borrowed without translation*
    
    | matsya (sa) | matsyalu (te) | fish |
    | pazha.m (ta) | phala (hi) | fruit |

- **Named Entities**
  - *do not change across languages*
    
    | mu.mbal (hi) | mu.mbal (pa) | mu.mbal (pa) |
    | keral (hi) | k.eraLA (ml) | keraL (mr) |

- **Fixed Expressions/Idioms**
  - *MWE with non-compositional semantics*
    
    | dAla me.n kuCha kAlA honA | (hi) | Something fishy |
    | dALa mA kAlka kALu hovu | (gu) |
We want to similar sentences to have similar embeddings.

We will find more matches at the sub-word level.

Can we use subwords as representation units?

Which subword should we use?
Simple Units of Text Representation

**Transliterate unknown words** [Durrani, et al. (2010), Nakov & Tiedemann (2012)]
(a) Primarily used to handle proper nouns   (b) Limited use of lexical similarity

स्वातंत्र्य  →  स्वतंत्रता
Translation of shared lexically similar words can be seen as kind of transliteration

**Character** [Vilar, et al. (2007), Tiedemann (2009)]
Limited context of character level representation

Character n-gram ⇒ increase in data sparsity

Limited benefit ....
... just for closely related languages

Macedonian - Bulgarian, Hindi-Punjabi, etc.
Orthographic Syllable  *(Kunchukuttan & Bhattacharyya, 2016a)*

(CONSONANT) + VOWEL

**Examples:** ca, cae, coo, cra, की (kI), प्रे (pre)

अभिमान ➔ अभिमान

**Pseudo-Syllable**

True Syllable ⇒ Onset, Nucleus and Coda
Orthographic Syllable ⇒ Onset, Nucleus

- Generalization of *akshara*, the fundamental organizing principle of Indian scripts
- Linguistically motivated, variable length unit
- *Number of syllables in a language is finite*
- Used successfully in transliteration
Byte Pair Encoded (BPE) Unit

(Kunchukuttan & Bhattacharyya, 2017a; Nguyen and Chang, 2017)

- There may be frequent subsequences in text other than syllables
- Herdan-Heap Law $\Rightarrow$ Syllables are not sufficient
- These subsequences may not be valid linguistic units
- But they represent statistically important patterns in text

How do we identify such frequent patterns?

Byte Pair Encoding (Sennrich et al, 2016), Wordpieces ( Wu et al, 2016), Huffman encoding based units (Chitnis & DeNero, 2015)
Byte Pair Encoded (BPE) Unit

Byte Pair Encoding is a compression technique (Gage, 1994)

Number of BPE merge operations=3
Vocab: A B C D E F

Data-dependent segmentation

- Inspired from compression theory
- MDL Principle (Rissanen, 1978) ⇒ Select segmentation which maximizes data likelihood
Example of various translation units

<table>
<thead>
<tr>
<th>Basic Unit</th>
<th>Symbol</th>
<th>Example</th>
<th>Transliteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word</td>
<td>W</td>
<td>घरासमोरचा</td>
<td>gharAsamoracA</td>
</tr>
<tr>
<td>Morph Segment</td>
<td>M</td>
<td>घरा समोर चा</td>
<td>gharA samora cA</td>
</tr>
<tr>
<td>Orthographic Syllable</td>
<td>O</td>
<td>घ रा स मो र चा</td>
<td>gha rA sa mo racA</td>
</tr>
<tr>
<td>Character unigram</td>
<td>C</td>
<td>घरा समोरचा</td>
<td>gha r A sa m o ra c A</td>
</tr>
</tbody>
</table>

**something that is in front of home:** ghar=home, samora=front, cA=of

Various translation units for a Marathi word
Instead of a sequence of words, the input to the network is a sequence of subword units.
## Neural Machine Translation

*(Nguyen and Chang, 2017)*

<table>
<thead>
<tr>
<th></th>
<th>baseline</th>
<th></th>
<th>transfer</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BLEU</td>
<td>size</td>
<td>BLEU</td>
<td>size</td>
</tr>
<tr>
<td><strong>Tur-Eng</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>word-based</td>
<td>8.1</td>
<td>30k</td>
<td>8.5*</td>
<td>30k</td>
</tr>
<tr>
<td>BPE</td>
<td>12.4</td>
<td>10k</td>
<td>13.2*</td>
<td>20k</td>
</tr>
<tr>
<td><strong>Uyg-Eng</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>word-based</td>
<td>8.5</td>
<td>15k</td>
<td>10.6\†</td>
<td>15k</td>
</tr>
<tr>
<td>BPE</td>
<td>11.1</td>
<td>10k</td>
<td>15.4\‡</td>
<td>8k</td>
</tr>
</tbody>
</table>

Uzbek as resource-rich assisting language; Turkish and Uyghur as primary languages
Size: refers to vocabulary size
Statistical Machine Translation

(Kunchukuttan & Bhattacharyya, 2016a; Kunchukuttan & Bhattacharyya, 2017a)

- Substantial improvement over char-level model (27% & 32% for OS and BPE resp.)
- Significant improvement over word and morph level baselines (11-14% and 5-10% resp)
- Improvement even when languages don't belong to same family (contact exists)
- More beneficial when languages are morphologically rich

<table>
<thead>
<tr>
<th>Src-Tgt</th>
<th>Char</th>
<th>Word</th>
<th>Morph</th>
<th>OS</th>
<th>BPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ben-hin</td>
<td>27.95</td>
<td>32.47</td>
<td>32.17</td>
<td><strong>33.54</strong></td>
<td>33.22</td>
</tr>
<tr>
<td>pan-hin</td>
<td>71.26</td>
<td>70.07</td>
<td>71.29</td>
<td><strong>72.41</strong></td>
<td>72.22</td>
</tr>
<tr>
<td>kok-mar</td>
<td>19.83</td>
<td>21.30</td>
<td>22.81</td>
<td>23.43</td>
<td><strong>23.63</strong></td>
</tr>
<tr>
<td>mal-tam</td>
<td>4.50</td>
<td>6.38</td>
<td>7.61</td>
<td>7.84</td>
<td><strong>8.67</strong>†</td>
</tr>
<tr>
<td>tel-mal</td>
<td>6.00</td>
<td>6.78</td>
<td>7.86</td>
<td>8.50</td>
<td><strong>8.79</strong></td>
</tr>
<tr>
<td>hin-mal</td>
<td>6.28</td>
<td>8.55</td>
<td>9.23</td>
<td>10.46</td>
<td><strong>10.73</strong></td>
</tr>
<tr>
<td>mal-hin</td>
<td>12.33</td>
<td>15.18</td>
<td>17.08</td>
<td>18.44</td>
<td><strong>20.54</strong></td>
</tr>
<tr>
<td>bul-mac</td>
<td>20.61</td>
<td>21.20</td>
<td>-</td>
<td><strong>21.95</strong></td>
<td>21.73</td>
</tr>
<tr>
<td>dan-swe</td>
<td>35.36</td>
<td>35.13</td>
<td>-</td>
<td>35.46</td>
<td><strong>35.77</strong></td>
</tr>
<tr>
<td>may-ind</td>
<td>60.50</td>
<td><strong>61.33</strong></td>
<td>-</td>
<td>60.79</td>
<td>59.54†</td>
</tr>
</tbody>
</table>
Named Entity Recognition

(Rudramurthy et al., 2018)

<table>
<thead>
<tr>
<th>Approach</th>
<th>Tamil</th>
<th>Malayalam</th>
<th>Bengali</th>
<th>Marathi</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRF + POS</td>
<td>44.60</td>
<td>48.70</td>
<td>52.44</td>
<td>44.94</td>
</tr>
<tr>
<td>CNN Bi-LSTM</td>
<td>52.34</td>
<td>55.37</td>
<td>50.34</td>
<td>56.53</td>
</tr>
<tr>
<td>CNN Bi-LSTM + Sub-word</td>
<td>52.34</td>
<td>56.82</td>
<td>52.56</td>
<td>50.25</td>
</tr>
<tr>
<td>CNN Bi-LSTM All</td>
<td>53.47</td>
<td>56.75</td>
<td>53.90</td>
<td>57.37</td>
</tr>
</tbody>
</table>
Utilizing Syntactic Similarity

(Kunchukuttan et al., 2014)

Phrase based MT is not good at learning word ordering

Solution: Let’s help PB-SMT with some preprocessing of the input

Change order of words in input sentence to match order of the words in the target language

Let’s take an example

Bahubali earned more than 1500 crore rupee sat the boxoffice
Parse the sentence to understand its syntactic structure

Apply rules to transform the tree

\[ VP \rightarrow \text{VBD} \ NP \ PP \Rightarrow VP \rightarrow PP \ NP \ VBD \]

This rule captures Subject-Verb-Object to Subject-Object-Verb divergence
Prepositions in English become postpositions in Hindi

\[ PP \rightarrow \text{IN NP} \Rightarrow PP \rightarrow \text{NP IN} \]

The new input to the machine translation system is

*Bahubali the boxoffice at 1500 crore rupees earned*

Now we can translate with little reordering

बाहुबली ने बॉक्सओफिस पर 1500 करोड रुपए कमाए

These rules can be written manually or learnt from parse trees
Can we reuse English-Hindi rules for English-Indian languages?

All Indian languages have the same basic word order

<table>
<thead>
<tr>
<th></th>
<th>Indo-Aryan</th>
<th>Dravidian</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>pan</td>
<td>hin</td>
</tr>
<tr>
<td>Baseline</td>
<td>15.83</td>
<td>21.98</td>
</tr>
<tr>
<td>Generic</td>
<td>17.06</td>
<td>23.70</td>
</tr>
<tr>
<td>Hindi-tuned</td>
<td>17.96</td>
<td>24.45</td>
</tr>
</tbody>
</table>

(Kunchukuttan et al., 2014)

**Generic reordering** (Ramanathan et al 2008)
Basic reordering transformation for English→ Indian language translation

**Hindi-tuned reordering** (Patel et al 2013)
Improvement over the basic rules by analyzing English → Hindi translation output
Utilizing Orthographic Similarity

*Orthographically Similar Languages*

(a) *highly overlapping phoneme sets*

(b) *mutually compatible orthographic systems*

(c) *similar grapheme to phoneme mappings*

*Can be useful in multilingual settings like:*

*Transliteration, grapheme to phoneme, Speech recognition, TTS, short text translation for related languages (tweets, headlines),*
Multilingual Neural Transliteration (Kunchukuttan et al., 2018)

Compact Architecture
- Shared embeddings, encoder, decoder and attention layer

Language Specific Output Layer

Compact Network

Shared LSTM Decoder
- context vector
  - previous state & output

Shared CNN Encoder
- annotation vectors

Shared Character Embedding Layer

L1 Output Layer

L2 Output Layer

Language

T E N D U L K A R

T E N D U L K A R
Top-1 accuracy for Phrase-based (P), bilingual neural (B) and multilingual neural (P)

**Major reduction in vowel related errors**

**Reduction in confusion between similar consonants**

*e.g. (T,D), (P,B)*

**Generates more canonical outputs**

*For मोरिस, moris is a valid spelling but maurice is canonical*
Why does Multilingual Training help?

Encoder learns specialized contextual representations
Outline

• Learning Cross-lingual Embeddings

• Training a Multilingual NLP Application

• Related Languages and Multilingual Learning

• Summary and Research Directions
Summary

• Cross-lingual word embeddings are the cornerstone for sharing training data across languages
• Tremendous advances in unsupervised learning of cross-lingual embeddings
• Ensuring word embeddings map to a common space is not sufficient
  • Encoder outputs have to be mapped too
• Related languages can make maximum utilization of task similarity and share data
Research Directions

• Do cross-lingual embeddings work equally well for all languages?
• Cross-lingual contextualized embedding \textit{i.e.} encoder outputs
• Alternative architectures
  • Transformer architecture shown to work better for multilingual NMT
  • Adversarial learning looks promising
• Target side sharing of parameters is under-investigated
Other Reading Material

• Tutorial on *Multilingual Multimodal Language Processing Using Neural Networks*. Mitesh Khapra and Sarath Chandar. NAACL 2016.


• Tutorial on *Statistical Machine Translation for Related languages*. Pushpak Bhattacharyya, Mitesh Khapra, Anoop Kunchukuttan. NAACL 2016.

• Tutorial on *Statistical Machine Translation and Transliteration for Related languages*. Mitesh Khapra, Anoop Kunchukuttan. ICON 2015.
Tools

• Multilingual Unsupervised and Supervised Embeddings (MUSE)
• VecMap

Thank you!

Multilingual data, code for Indian languages


http://www.cfilt.iitb.ac.in

https://www.cse.iitb.ac.in/~anoopk
References

References


Kunchukuttan, A., & Bhattacharyya, P. (2016a). Orthographic syllable as basic unit for smt between related languages. EMNLP.


Kunchukuttan, A., & Bhattacharyya, P. (2017a). Learning variable length units for SMT between related languages via Byte Pair Encoding. SCLeM.


References

References


