Unsupervised Transliteration Mining with Application to Statistical Machine Translation

Nadir Durrani

Qatar Computing Research Institute

Transliteration

- Languages are written in different scripts
  - Russian, Bulgarian and Serbian – written in Cyrillic Script
  - Urdu, Farsi and Pashto – written in Arabic Script
  - Hindi, Marathi and Nepalese – written in Devanagri

- Transliteration is converting text in one script into another
  - Pronunciation of words remain roughly the same
  - талботу (tælbət) → Talbot
  - مورغان (morghan) → Morgan
  - सीमा (sima) → Seema

Utility

- Transliteration can benefit major NLP applications
  - Cross language information retrieval
  - Terminology extraction
  - Machine translation
    - Translation of OOV words
    - Learning when to transliterate (Hermjakob 2008; Azab 2013)
      - E.g. “Dudley North visits North London”
    - Translating closely related languages (Nakov and Tiedeman 2012)
      - E.g. Bulgarian/Macedonian, Thai/Lao, Hindi/Urdu

Building a Transliteration System

- Rule-based approach
  - Manually built transliteration rules ح/ہ/ھ → h, ک/ق → q,k,c
  - Use edit-distance based techniques to score variants
  - Problem: Linguistic knowledge + effort required
- Data-driven approach
  - Learns transliteration rules automatically from the data
  - Problem: Requires a list of transliteration pairs

Transliteration Mining

- Solution: Mine transliteration pairs from parallel data

دوکان	Shop
طاقت	Power
نعمت	Blessing
ایڈنبرگ	Edinburgh
یونیورسٹی	University
ضرورت	Need
...	...
...	...
ریسرچ	Research

Transliteration Corpus

ایڈنبرگ	Edinburgh
یونیورسٹی	University
ریسرچ	Research

Approaches to Transliteration Mining

- Supervised and Semi-Supervised Approach
  - Sherif and Kondrak, 2007; Kahki et. al., 2011; Jiampojamarn et al., 2010; Noeman and Madkour, 2010

- Unsupervised Approach
  - Sajjad et al., 2012 (Fully unsupervised)
    - Based on EM algorithm

Unsupervised Transliteration Mining

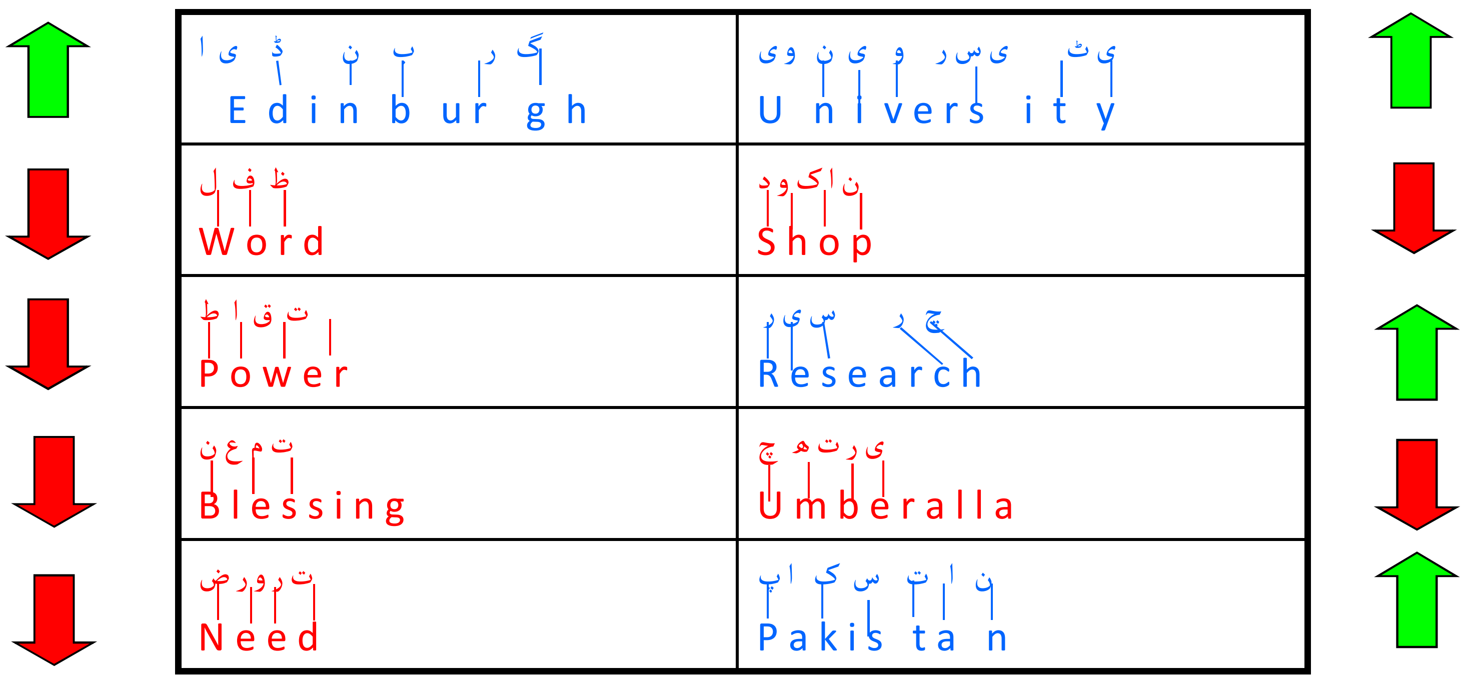
- Basic Idea
  - If we have a transliteration model, we can score the training data to extract transliteration corpus

а н а л о г a n a l o g	0.83
с и с т е м а a n a l o g	0.05
э н т о н и a n t h o n y	0.71
я з ы к о в о l i n g u i s t i c	0.001

а | a	0.78
э| a	0.45
а | e	0.07
г| g	0.75
и| y	0.88
л| l	0.82
…	…
…	…

Unsupervised Transliteration Mining

- Basic Idea
  - If we have a transliteration model, we can score the training data to extract transliteration corpus
  - If we knew which pairs in the training data are transliterations we can build transliteration model from these/boast these pairs



Unsupervised Transliteration Mining

- Transliteration Model
  - Joint sequence model
  - Only 1-1/1-ε/ε-1 mappings
  - No reoredering
  - Independence assumption
  - Sums over all character alignment sequences "a" of a word pair

$$p_1(e,f) = \sum_{a \in Align(e,f)} \prod_{j=1}^{|a|} p(q_j)$$

ا ی ڈ ε ن ب ε ر گ ε ε E d i n b u r g h	q_1: ا-εq_2: ی-E q_3: ڈ-d q_4: ε-i ……q_9: گ-g q_{10}:ε- h
ا ی ڈ ε ن ب ε ر ε گ E ε d i n b u r g h	q_1: ا-E q_2: ی-εq_3: ڈ-d q_4: ε-i ……q_9: ε- g q_{10} گ- h

Two different alignment sequences of a word pair

Unsupervised Transliteration Mining

- Overall model
  - We want EM to maximize the likelihood the entire training data
  - Transliteration model should only model the transliteration sub-data

$$p_1(e,f) = \sum_{a \in Align(e,f)} \prod_{j=1}^{|a|} p(q_j) \qquad p_2(e,f) = \prod_{i=1}^{|e|} p_E(e_i) \prod_{i=1}^{|f|} p_F(f_i)$$

Transliteration Model Non-Transliteration Model

  - A mixture of transliteration and non-transliteration model

$$p(e,f) = (1-\lambda)p_1(e,f) + \lambda p_2(e,f)$$

  - Posterior Probability

$$\frac{(1-\lambda)p_1(e,f)}{p(e,f)} \qquad \frac{\lambda p_2(e_i,f_i)}{p(e_i,f_i)}$$

Transliteration Model Non-Transliteration Model

Unsupervised Transliteration Mining

Expectation Step

- Compute expected counts for all bilingual character pairs “q”

$$c(q) = \sum_{i=1}^{N} \sum_{a \in Align(e_i, f_i)} \frac{(1-\lambda) p_1(a, e_i, f_i)}{p(e_i, f_i)} n_q(a)$$

$$c_{ntr} = \sum_{i=1}^{N} \frac{\lambda p_2(e_i, f_i)}{p(e_i, f_i)}$$

λ= prior probability of non-transliteration
$p_1(a,e_i,f_i)$ = probability of an alignment sequence “a”
$n_q(a)$ = number of times “q” occurs in “a”
c_{ntr} = sum of non-transliteration posterior probabilties

Unsupervised Transliteration Mining

- Maximization Steps

$$p(q) = \frac{c(q)}{\sum_{q'} c(q')} \qquad \lambda = \frac{c_{ntr}}{N}$$

λ= prior probability of non-transliteration
c_{ntr} = sum of non-transliteration posterior probabilities
p(q) = probability of a bilingual unit “q”

Intrinsic Evaluation

- Shared Task of Transliteration Mining (Kumaran et al. 2010)
  - Mine transliterations from a list of word pairs
  - Comparing F-Measures against best submitted system

Language	Unsupervised Mining	Best System
Arabic	P: 89.2 R: 95.7 F: 92.4	F:91.5
Hindi	P: 92.6 R: 99 F: 95.7	F:94.4
Russian	P: 67.2 R: 97.1 F: 79.4	F:87.5

Integration into Machine Translation

- Run unsupervised transliteration over word-alignments
  - 7 Language pairs:
    - Arabic, Bengali, Farsi, Hindi, Russian, Telegu and Urdu
  - Only 1-1 alignments are used as N-1/M-N alignments are less likely to be transliterations
  - Output: List of transliteration pairs

- Build transliteration model
  - We use phrase-based Moses
    - Segment training data into characters
    - 4-translation features
    - Monotonic decoding
  - Use 10% training data for tuning parameters

Evaluation

Lang	Data	$Train_{tm}$	$Train_{tr}$	Dev	$Test_1$	$Test_2$
		Sent	Types			
Arabic	IWSLT-13	152K	6795	887	1434	1704
Bengali	JHU	24K	1916	775	1000	
Farsi	IWSLT-13	79K	4039	852	1185	1116
Hindi	JHU	39K	4719	1000	1000	
	WMT-14	275K	38K	500	2500	
Russian	WMT-13	2M	302K	1501	1502	3000
Telugu	JHU	45K	4924	1000	1000	
Urdu	JHU	87K	9131	980	883	

Integration into Machine Translation

- Run unsupervised transliteration over word-alignments
  - Only 1-1 alignments are used as N-1/M-N alignments are less likely to be transliterations
  - Output: List of transliteration pairs
- Build transliteration model
  - We use phrase-based Moses
    - Segment training data into characters
    - 4-translation features
    - Language model trained on target-side
    - Monotonic decoding
  - Use 10% training data for tuning parameters

Intrinsic Evaluation

Accuracy	AR	HI	RU
Test Size	1799	2394	1859
1-best	20.0%	25.3%	46.1%
100-best	80.2%	79.3%	87.5%

- Test Data = Seed Data + Reference Data provided for Transliteration Mining Shared Task (Kumaran et al. 2010)
- 1-best accuracy is quite low
- But 100-best accuracy is reasonable
- Hopefully MT system will bring out MT system at the top

Integration into Machine Translation

- Three methods for integration
  - Method 1: Replace OOV words with 1-best transliteration
  - Method 2: Selects transliteration from n-best list in post-decoding
  - Method 3: Integrates transliteration phrase-table inside decoder

Integration into Machine Translation

- Three methods for integration
  - Method 1: Replace OOV words with 1-best transliteration
    - Does not consider contextual information,
    - بيل → "Bell" in "Alexander Graham Bell"
    - بيل → "Bill" in "Bill Clinton"

SMT Evaluation

Lang	Test	B_0	M_1	M_2	M_3	OOV
AR	$iwslt_{11}$	26.75	+0.12	+0.36	+0.25	587
	$iwslt_{12}$	29.03	+0.10	+0.30	+0.27	682
BN	jhu	16.29	+0.12	+0.42	+0.46	1239
FA	$iwslt_{11}$	20.85	+0.10	+0.40	+0.31	559
	$iwslt_{12}$	16.26	+0.04	+0.20	+0.26	400
HI	jhu	15.64	+0.21	+0.35	+0.47	1629
RU	wmt_{12}	33.95	+0.24	+0.55	+0.49	434
	wmt_{13}	25.98	+0.25	+0.40	+0.23	799
TE	jhu	11.04	-0.09	+0.40	+0.75	2343
UR	jhu	23.25	+0.24	+0.54	+0.60	827
Avg		21.9	+0.13	+0.39	+0.41	950

Integration into Machine Translation

- Three methods for integration
  - Method 1: Replace OOV words with 1-best transliteration
    - Does not consider contextual information,
    - بيل → "Bell" in "Alexander Graham Bell"
    - بيل → "Bill" in "Bill Clinton"
  - Method 2: Selecting the best transliteration from a list of n-best transliteration in a post-decoding step
    - Pipe the output of decoder into monotonic decoder
    - Features: Language Model, LM-OOV feature, Transliteration Phrase Table
    - 4 translation features to form a transliteration phrase-table

	Bill	
Alexander Graham	Bell	is credited with the invention of telephone
	Ball	
	Pill	

SMT Evaluation

Lang	Test	B_0	M_1	M_2	M_3	OOV
AR	$iwslt_{11}$	26.75	+0.12	+0.36	+0.25	587
	$iwslt_{12}$	29.03	+0.10	+0.30	+0.27	682
BN	jhu	16.29	+0.12	+0.42	+0.46	1239
FA	$iwslt_{11}$	20.85	+0.10	+0.40	+0.31	559
	$iwslt_{12}$	16.26	+0.04	+0.20	+0.26	400
HI-EN	jhu	15.64	+0.21	+0.35	+0.47	1629
	wmt_{14}	14.67		**+0.81**		503
EN-HI	wmt_{14}	11.76		**+1.07**		394
RU	wmt_{12}	33.95	+0.24	+0.55	+0.49	434
	wmt_{13}	25.98	+0.25	+0.40	+0.23	799
TE	jhu	11.04	-0.09	+0.40	+0.75	2343
UR	jhu	23.25	+0.24	+0.54	+0.60	827

Integration into Machine Translation

- Three methods for integration
  - Method 2: can not reorder unknown words
    - عرب بحيره
    (Arabian Sea) instead translates to Sea Arabian
  - Method 3 is also useful when translating words that can also be transliterated
    - आशा (Asha) translates into "hope" but transliterates to "Asha" in "Asha Bhosle" (the famous Indian singer)
    - Learning what to transliterate all previous work is language dependent
  - Method 3: Passes transliteration phrase-table into the decoder
    - Transliteration phrase-table
    - All features + LM-OOV feature

SMT Evaluation

Lang	Test	B_0	M_1	M_2	M_3	OOV
AR	$iwslt_{11}$	26.75	+0.12	+0.36	+0.25	587
	$iwslt_{12}$	29.03	+0.10	+0.30	+0.27	682
BN	jhu	16.29	+0.12	+0.42	+0.46	1239
FA	$iwslt_{11}$	20.85	+0.10	+0.40	+0.31	559
	$iwslt_{12}$	16.26	+0.04	+0.20	+0.26	400
HI	jhu	15.64	+0.21	+0.35	+0.47	1629
RU	wmt_{12}	33.95	+0.24	+0.55	+0.49	434
	wmt_{13}	25.98	+0.25	+0.40	+0.23	799
TE	jhu	11.04	-0.09	+0.40	**+0.75**	2343
UR	jhu	23.25	+0.24	+0.54	+0.60	827
Avg		21.9	+0.13	+0.39	+0.41	950

SMT Evaluation

- Can we improve these results by improving 1-best accuracy?
  - Replace mined transliteration system (MTS) with gold-standard transliteration system (GST)

	AR		HI	RU	
Test	$iwslt_{11}$	$iwslt_{12}$	jhu	wmt_{12}	wmt_{13}
B_0	26.75	29.03	15.64	33.95	25.98
MTS	27.11	29.33	16.11	34.50	26.38
GST	26.99	29.20	16.11	34.33	26.22
Δ	-0.12	-0.13	0.0	-0.17	-0.16
	Transliteration Pairs Used				
MTS	6795		4719	302K	
GST	1799		2394	1859	

Error Analysis

- MTS has better rule coverage – GST suffers from data sparsity

<table>
<tr><th>Source</th><th>MTS/Ref</th><th>GST</th></tr>
<tr><td>الغيغابكسل</td><td>Gigapixel</td><td>algegapixel</td></tr>
<tr><td colspan="3">ال (al) → ε</td></tr>
<tr><td>سبرلوك</td><td>Spurlock</td><td>Sbrlok</td></tr>
<tr><td colspan="3">ب (b) → p</td></tr>
<tr><td>талботу</td><td>Talbot</td><td>Talboty</td></tr>
<tr><td colspan="3">y → ε</td></tr>
</table>

Urdu → Hindi-to-English Machine Translation

- Huge Vocabulary Overlap
  - Small study on Hindustan times and Dianik Jagran
  - 40% Hindi types overlap with Urdu
- Hindi-Urdu Machine Translation is fairly trivial
  - Same grammatical structure
  - Written in different scripts
  - NLP resources in one language can be used in other
- Construct Urdu – Hindi phrase table by interpolating Urdu – English and English – Hindi phrase-tables
  - Construct Transliteration Phrase-table by running EM algorithm over Urdu –Hindi Phrase-table

Urdu → Hindi-to-English Machine Translation

- Urdu → Hindi SMT system
  - EMILLE corpus (5K sentences)
  - Interpolated phrase-table
  - Transliteration phrase-table

System	Tune	Test
Baseline	34.01	34.64
+Transliteration	34.77	35.76
+Interpolation	38.00	37.99

- Hindi → English → Hindi system (WMT-14)

System	Dev	Test
Hi-EN	17.11	15.77
	17.60 Δ +0.49	15.97 Δ+0.20
EN-Hi	11.74	11.57
	12.83 Δ +1.09	12.47 Δ+0.90

Summary

- Unsupervised Transliteration Mining
- Integration into Moses
  - 3 Methods of integration
  - Achieved average gain of 0.41 ranging from (0.23 — 1.01) across 7 language pairs
  - Mined transliterations provide better rule coverage than gold-standard transliterations
  - Helpful for closely related language pairs (Urdu→Hindi)
  - All code is available for use in Moses git-repository

References

Hassan Sajjad, Alexander Fraser, Helmut Schmid (2012). A Statistical Model for Unsupervised and Semi-supervised Transliteration Mining. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)

Nadir Durrani, Hassan Sajjad, Hieu Hoang, Philipp Koehn (2014). Integrating an Unsupervised Transliteration Model into Statistical Machine Translation. In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)

Nadir Durrani and Philipp Koehn (2014). Improving Machine Translation via Triangulation and Transliteration. In Proceedings of the 17th Annual Conference of the European Association for Machine Translation (EAMT).

Nadir Durrani, Hassan Sajjad, Alexander Fraser and Helmut Schmid (2010). Hindi-to-Urdu Machine Translation through Transliteration. In Proceedings of the 48th Annual Conference of the Association for Computational Linguistics, Uppsala, Sweden.