Bootstrapping Transliteration with Constrained Discovery for Low-Resource Languages

Shyam Upadhyay **Jordan Kodner** **Dan Roth**
University of Pennsylvania, Philadelphia PA, USA

Transliteration

Generation

Ravi and Knight (2009), Irvine et al. (2010)

ब्रसेल्स → brussels

Generate transliteration in a open-ended way. **(a transduction task)**

Discovery

Sproat et al. (2006), Klementiev and Roth (2008)

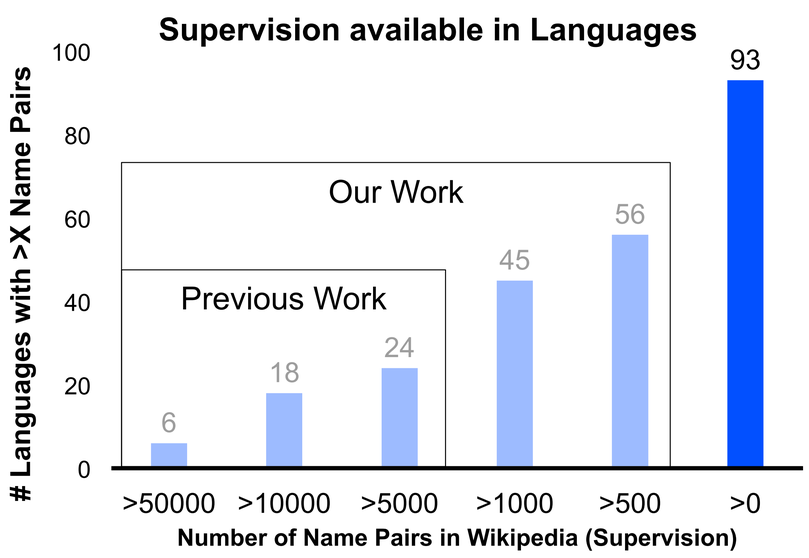
ब्रसेल्स → Name List: thomas, brussels, rupert

Select transliteration for a word from a (long) list of names. **(a ranking task)**

Our Work: Generation in low-resource settings.

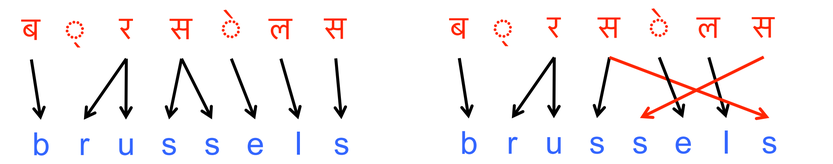
Idea: Discovery is a easier task. Use it to aid Generation.

Contributions of Our Work

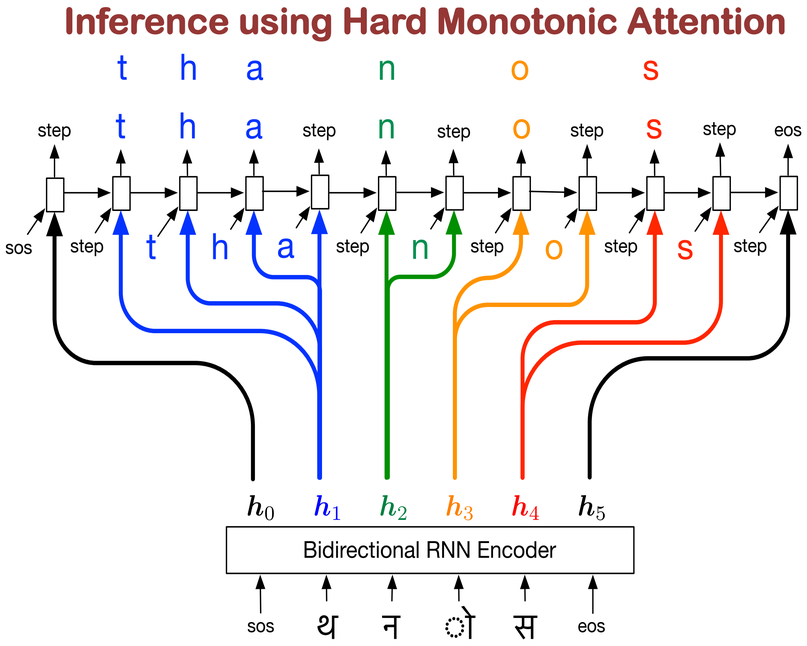


- A seq2seq generation model, tailored for transliteration.
- A bootstrapping algorithm, that uses constrained discovery to improve a weak generation model.

Transliteration as Monotonic Seq2Seq Generation



Inference using Hard Monotonic Attention



Encoding the Input

- The encoder encodes the character embeddings of the input characters using a bidirectional RNN.

Decoding with Hard Monotonic Attention

- The decoder generates a sequence of actions, where each action is either a character from the output alphabet, or a *step* action.
- At any time, the decoder RNN is attending on a **single** input character's hidden vector to generate output character(s).
- The *step* action increments the attention position by one.
- The stepping mechanism ensures that the decoding is **monotonic**.
- Inspired by Aharoni (2017)'s approach for morphological inflection.

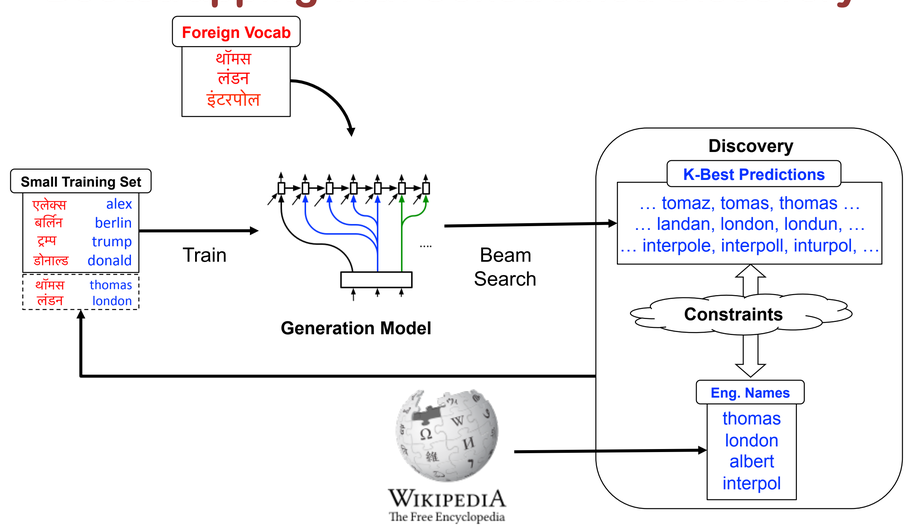
Training

The oracle sequence of actions is generated from name pairs using Algorithm 1 from Aharoni (2017), and the latent character-level alignments are derived using the algorithm from Cotterell (2016).

Inference Strategies

- **Unconstrained (U)** − pick the most likely transliteration from beam.
- **Dictionary Constrained (DC)** − pick the most likely transliteration from beam that appears in a name dictionary, else default to unconstrained strategy.

Bootstrapping with Constrained Discovery



After every iteration, purge the set of mined name pairs to prevent new model to be affected by (bad) pairs mined in earlier iterations.

Constraints

- Minimum length of exact match - False positives in early iterations were usually short transliterations.
- The length ratio of output string and input string should be close to ratio estimated from training data.

Convergence

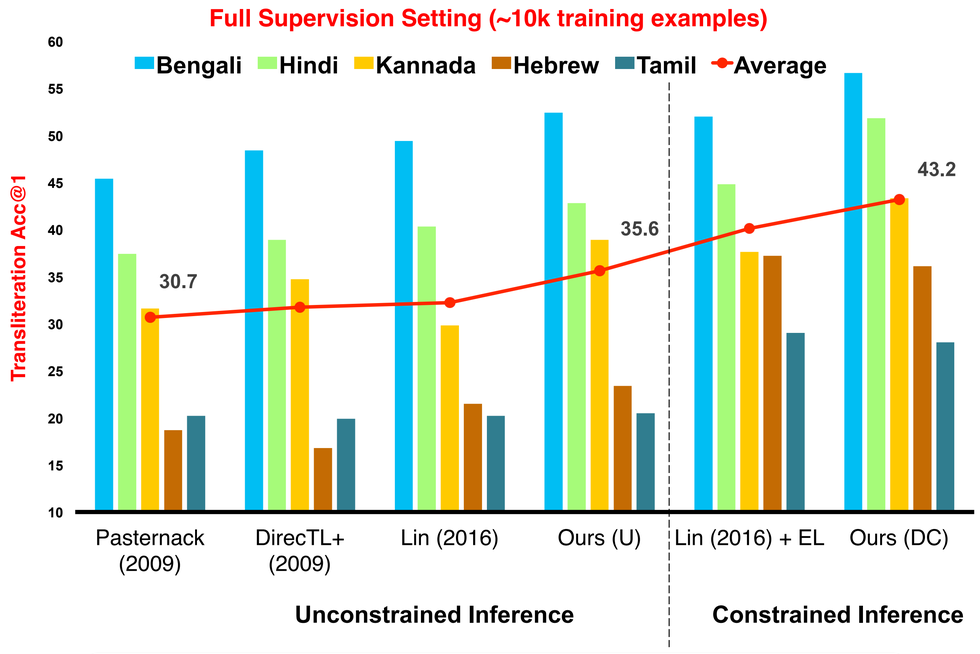
Keep bootstrapping until accuracy@1 stops increasing on dev set.

Full Supervision Experiment

- **Evaluation Dataset:** NEWS 2015
- Each language has ~10k (or more) name pairs for supervision.

- **Models Compared**

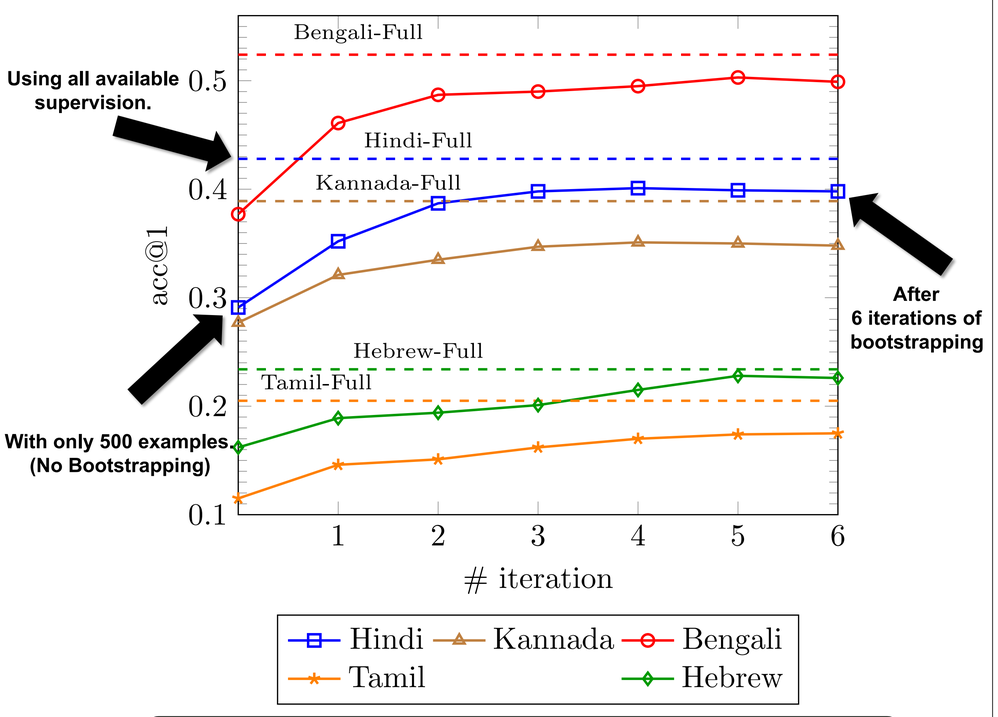
1. *Pasternack (2009)* − probabilistic transliteration approach that uses alignment b/w substrings in both source and target names.
2. *DirecTL+ (2009)* - HMM-like discriminative string transduction model.
3. *Lin (2016)* − A transliteration approach based on the joint source-channel model, that uses many-2-many alignments b/w source and target.
4. *Lin (2016) + EL* − re-rank transliterations lang-indep. entity linking.
5. *Ours(U & DC)* − unconstrained and dict. constrained version of our model.



Hard Monotonic Attention Model is better than SoTA.
Simple dictionary constrained inference, does much better than the expensive SoTA + Entity Linking approach

Low Resource Experiment

- Only 500 name pairs available in each language as supervision.
- Train a weak generation model and bootstrap using a name dict.



Starting with only 500 examples, bootstrapping achieves competitive performance to full supervision.

Inherent Challenges of Transliteration

Source Driven Errors

Tamil: {*ta*, *da*, *tha*, *dha*} → {த}
Hindi: {*ta, da, tha, dha*} → {त, द, थ, ध}

Acc@1	Hindi as Src	Hindi as Trg
Tamil	31%	15%

Target Driven Errors

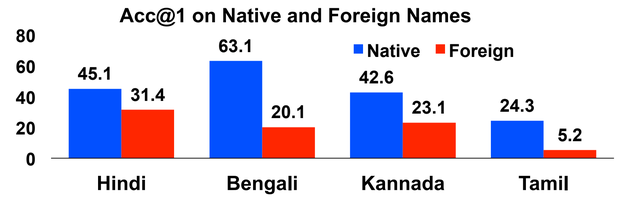
•Irregular Spelling
[Ph]iladel[ph]ia, So[ph]ia, [F]rance
[K]ansas, [C]ardiff, [Q]uinn, Bro[ck]

•Inconsistency with Devoicing
(Медведе[в], Medvede[v]), (Ивано[в], Ivano[v])
(Смирно[в], Smirno[ff]), (Рахманино[в], Rachmanino[ff])

•Silent Letters
Marsei[lle], Versai[lles], Bruxell[es]

Native vs Foreign Names

Transliterating *Wickramsinghe* vs *Brussels* from Tamil.



Manual Annotation Exercise

- **Languages:** Punjabi and Armenian.
- Two subtasks for each annotator.
- **Task 1:** Two annotations per letter ("[J]ulia", "Ben[j]amin")
- **Task 2:** Transliterate list of English words.

Lang.	Punjabi	Armenian
Time (hours)	5	4
Pairs	~500	~600
Ours (U)	33.4	49.9
Ours(U) + Boot.	44.5	55.8

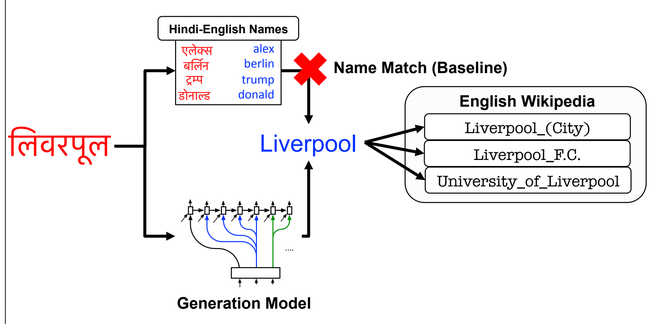
Manual annotation is practical and effective! Enough supervision to bootstrap a model.

Extrinsic Evaluation

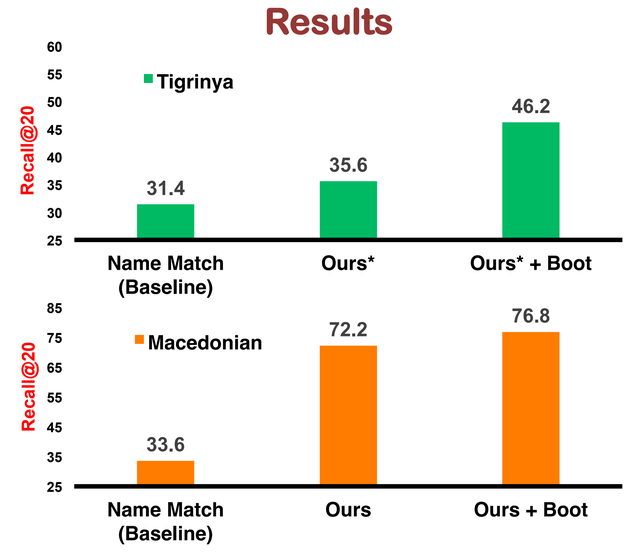
Task: Candidate Generation (CG) for cross-lingual entity linking.
Example: A mention of "Chicago" in Amharic is first transliterated from ሺካጎ, and then candidate entities are generated.

ሺካጎ በዎድስቶክ ይጫወታል.
(Chicago will play at Woodstock.)

Languages: Macedonian and Tigrinya
Evaluation Metric: if the gold entity for the query is in the top-20 candidates (Recall@20).



Results



References

Sproat (2006) Named Entity Transliteration with Comparable Corpora. Richard Sproat, Tao Tao, and ChengXiang Zhai. **COLING-ACL 2006.**
Pasternack (2009) Learning Better Transliterations. Jeff Pasternack and Dan Roth. **CIKM 2009**.
DirecTL (2009) DirecTL: A Language-Independent Approach to Transliteration. Sittichai Jiampojamarn, Aditya Bhargava, Qing Dou, Kenneth Dwyer, and Grzegorz Kondrak. **NEWS 2009**.
Ravi and Knight (2009) Learning Phoneme Mappings for Transliteration without Parallel Data. Sujith Ravi and Kevin Knight. **NAACL 2009.**
Irvine (2010) Transliterating from All Languages. Ann Irvine, Chris Callison-Burch, and Alexandre Klementiev. **AMTA 2010**.
Klementiev (2008) Named Entity Transliteration and Discovery in Multilingual Corpora. Alex Klementiev and Dan Roth. In **Learning Machine Translation 2008**.
Lin (2016) Leveraging Entity Linking and Related Language Projection to Improve Name Transliteration. Ying Lin, Xiaoman Pan, Aliya Deri, Heng Ji, and Kevin Knight. **NEWS 2016**.
Cotterell (2016). The SIGMORPHON Shared Task - Morphological Reinflection. Ryan Cotterell, Christo Kirov, John Sylak-Glassman, David Yarowsky, Jason Eisner, Mans Hulden. **2016**.
Aharoni (2017). Morphological Inflection Generation with Hard Monotonic Attention. Roee Aharoni, Yoav Goldberg. **ACL 2017**.

This work was supported under DARPA LORELEI by Contract HR0011-15-2-0025, Agreement HR0011-15-2-0023 with DARPA, and an NDSEG fellowship for the second author.