



“Drinking Maicaraosoft...”

Talk on Language Neutral
Statistical Machine
Transliteration

by Tobias Kellner

ပြည်ထောင်စု
ဘီယို

ประเทศไทย

||T||A||L||

လောင်းလောင်း

لندن

இலண்

लंदन

ഇന്ത്യ



On work carried out in summer 2006 with the
Multilingual Systems Group
at **Microsoft Research India, Bangalore**

Supervised by A. Kumaran <a.kumaran@microsoft.com>

বৈদ্য শিক্ষা
কলকাত্তা

ประเทศไทย

||T||
لندن

இலண்டன்

ഇന്ത്യ

मेरा नाम तोवि है

my name तोवि is



Outline

- Introduction to Transliteration
 - What are we trying to do?
- Machine Transliterations
 - How can computers transliterate?
 - What is our approach?
- Results obtained
 - Does it work ?!?

What is Transliteration?

- Definition (short):
“Converting from one orthography to another”
- Definition (longer)
“Replacing a word in the source language/script with a phonetically identical word in the target language/script”

Transliteration: Example

नमस्ते → Namaste

is a transliteration, whereas

नमस्ते → Hello

is a translation

PUREVEG

ಹೊಸದೇಲಿ ಸಿಟಿ

Hotels Surabhi

होटल सु

ಹೊಸದೇಲಿ ಸಿಟಿ



Transliteration in MT

■ Out-Of-Vocabulary Words

- many (most?) OOV words are named entities
- Named Entities should be transliterated

■ Text alignment

- identifying cognates (“obvious translations”) for alignment of words and sentences

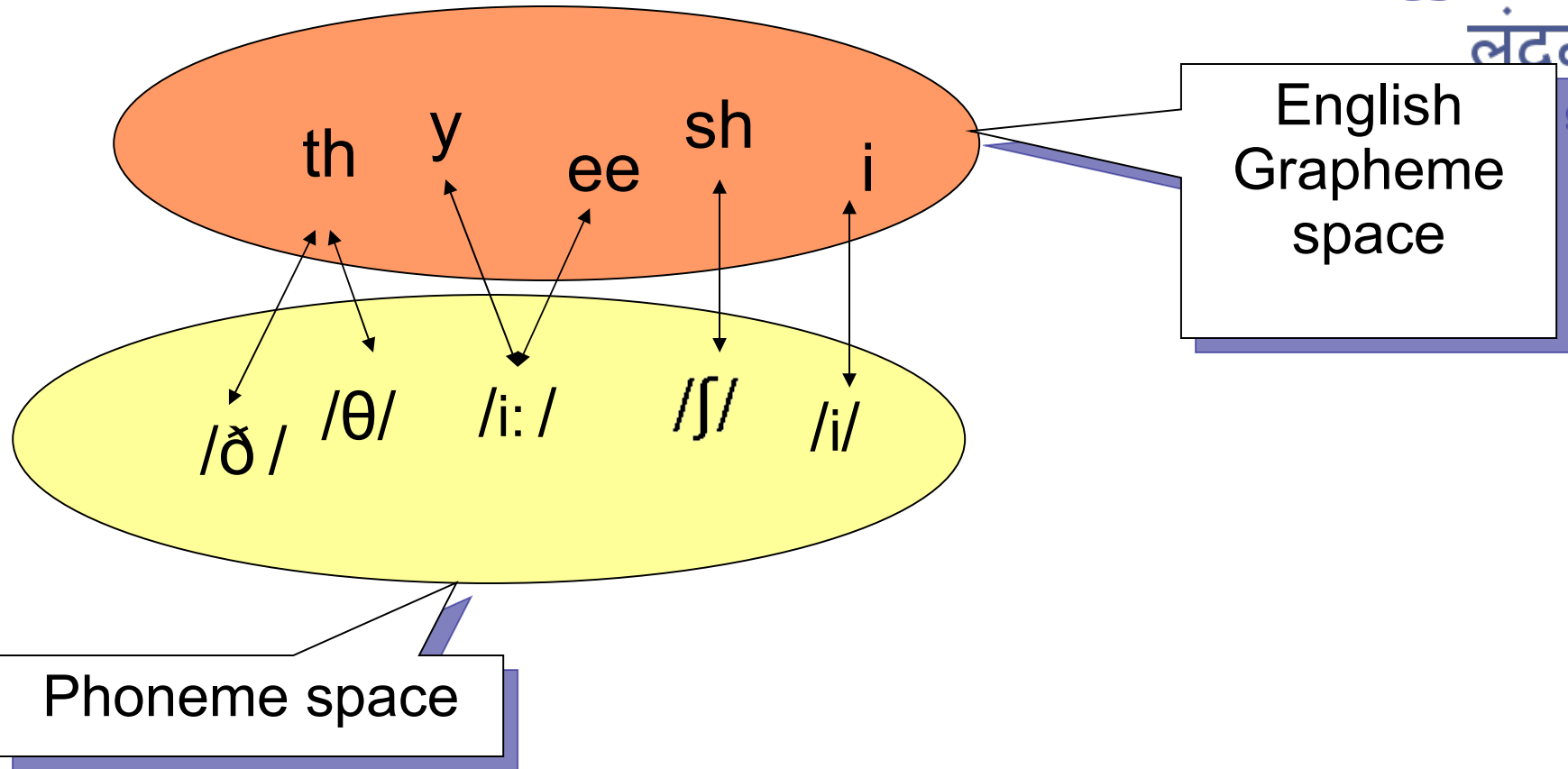
“Iraq's Foreign Minister Hoshyar Zebari said...”

“उधर इराक के विदेश मंत्री होशियार ज़ेबारी ..”

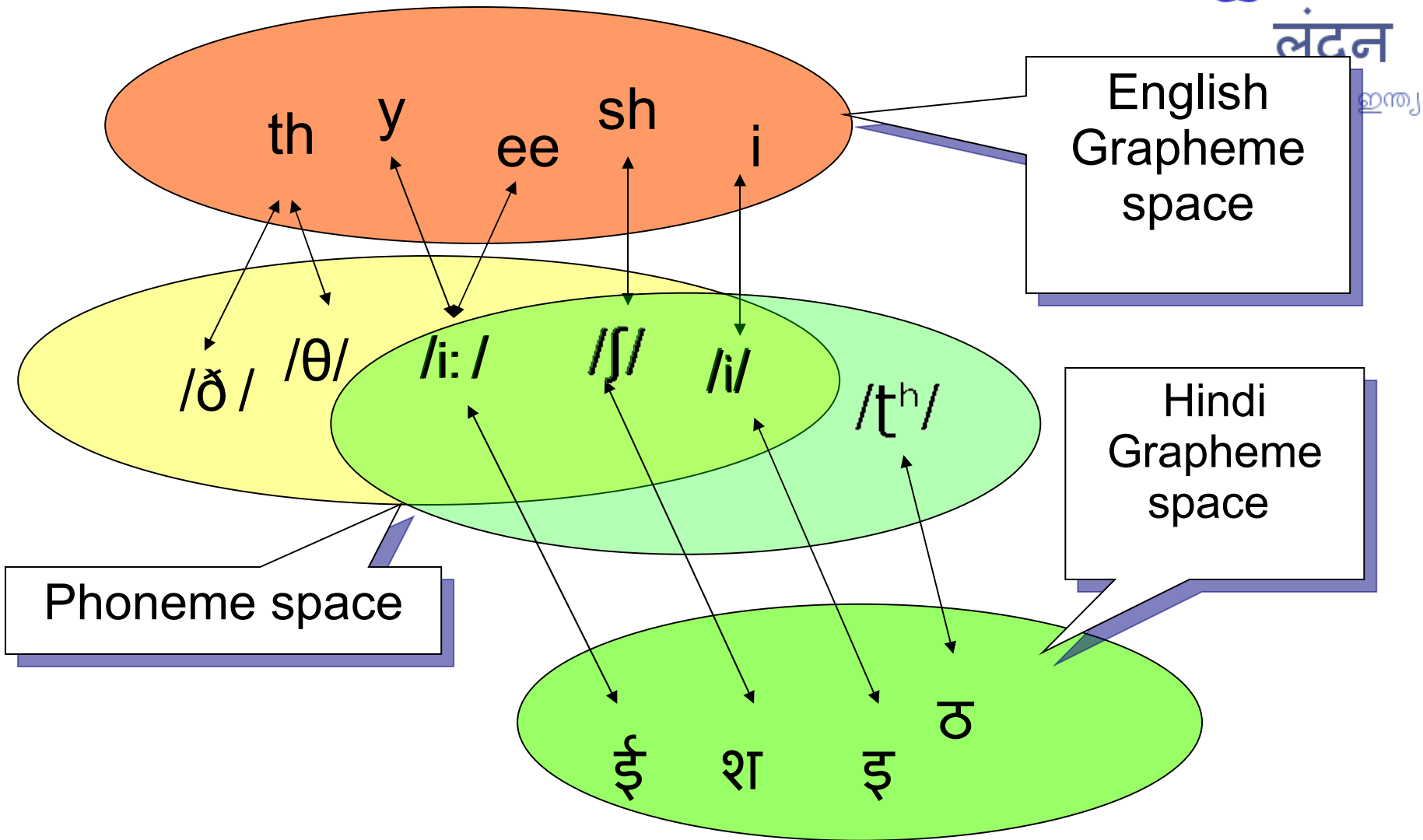
“udhar irak ke videsh mentri hoshiyar zebari...”

A closer look at the problem

A closer look at language



A closer look at language



Direction of transliteration

■ Forward Transliteration

- Word does not exist in Target Language
- results (usually) not in any dictionary/list
- (usually) all phonetically close results ok
- मीना → Mina
 - Minaa
 - Meena
 - Meenaa...

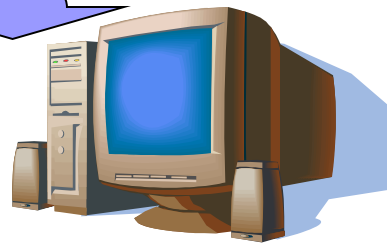
Direction of transliteration

■ Backward Transliteration

- Word exists in Target Language
- Correct form can be found in word lists
- Only one form is acceptable
- लंदन → London
 - *Lundan
 - *Lundon...

PART II: MACHINE TRANSLITERATION

नमस्ते



Namaste

Machine Transliteration

- most work on Chinese, Arabic, Korean, Katakana, Thai and Indic languages
- Three basic strategies:
 - 1) Manually compiled rules/tables (not discussed here)
 - 2) **Phonetic-based model** (“6-fold path”) (Knight & Graehl, 1998)
 - 3) **Direct Orthographical Mapping** (Al-Onaizan & Knight, 2002)

Source Word – “Athens”



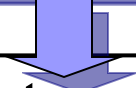
Source Segments – “a” “th” “e” “n” “s”



Source Phonemes – /æ/ /θ/ /ə/ /n/ /s/



Target Phonemes - /e/ /tʰ/ /e/ /n/ /s/



Target Segments – “ए” “थ” “ए” “ं” “स”



Target Word – “एथेस”

“The
six-fold
path”



Phonetic-based models

- initially: K. Knight, J. Graehl (1998):
“*Machine Transliteration*”
- several other papers for different languages, using
 - Hand-crafted Rules
 - Machine Learning from annotated data
- requires language-specific, linguistic knowledge, or data about segmentation & pronunciation

Direct Orthographic Modelling

- Y. Al-Onaizan, K. Knight (2002)
“Machine Translation of Names in Arabic Text”
- direct mapping between orthographic source- and target-segments
- Al-Onaizan & Knight: “[...] The spelling based model was by far more accurate than the phonetic-based model [...]”

Our strategy:

- Goals at MSRI: Transliteration which
 - is Language-neutral → no language-specific knowledge or rules required, “One code for all languages”
 - requires no annotated training data for segmentation or pronunciation
- strategy: Direct Orthographic Modelling with automated segmentation

	B	C	D	E
4686	Khirwadkar	खिरवडकर	கிர்வாத்கர்	வீர்வாட்கர்
4687	Ghule	घुले	குலே	஘ுலே
4688	Thipse	टिप्से	திப்ஸே	திப்சே
4689	Jambhale	जांभले	ஜம்பலே	ஜாம்பலே
4690	Mutatkar	मुटाटकर	முதாத்கர்	முதாத்கர்
4691	Tungaare	तुंगारे	துன்காரே	துன்காரே
4692	Urdhvareshhe	ऊध्वरेशे	உர்த்வரேஷே	உர்த்வரேஷே
4693	Telang	तेलंग	தெலன்க்	தெலன்க்
4694	Gawarikar	गवारीकर	கவாரிகர்	கவாரிகர்
4695	Raverkar	रावेरकर	ரவேர்கர்	ரவேர்கர்
4696	Yavalkar	यावलकर	யவல்கர்	யவல்கர்
4697	Kendurkar	केंदूरकर	கெந்துர்கர்	கெந்துர்கர்
4698	Hedgewar	हेडगेवार	ஹெட்கேவர்	ஹெட்கேவர்
4699	Wavde	वावदे	வாவதே	வாவதே
4700	Mande	मांडे	மண்டே	மண்டே
4701	Mendhe	मेंढे	மேந்தே	மேந்தே
4702	Indapurkar	इंदापुरकर	இந்தபுர்கார்	இந்தபுர்கார்

Our approach

- Uses “segments”: chunks of graphemes (usually representing one or more speech sounds)

e.g. “na”, “न”

- Expresses transliterations between a source- and a target-word as sequences of segment pairs:

Namaste → नमस्ते

as < न ,na> < म , ma> < स्त, st> < े,e>

- Calculates probabilities for transliterations between source- and target words based on transliteration probabilities for segment pairs

e.g. $P(\text{न} | \text{na})$

Our approach

- BUT:
 - What are the segments?
 - What are the probabilities?
- Solution: Training through iterative alignment (Viterbi Training)

Iterative Training

Generate Initial
Segment-Pairs



Training: Initialization

नमस्ते
namaste

- We only know two points: beginning of first segment, end of last segment
- Assumption: source- and target segments are at most n and m characters long
- We can create $n \times m$ segment pairs as hypotheses for first and last segment pair

Training: Initialization

नमस्ते



namaste

<n, न>, <n, नम>, <n, नमस> ,

<na, न>, <na, नम >, <na, नमस >

<nam, न>, <nam, नम >, <nam, नमस >

- do for all word pairs, count segment pairs
- Idea: Good pairs like <na,न> will have higher counts than artefacts like <n,नम>

Iterative Training

Generate Initial
Segment-Pairs

Compute
Segment-Pair-
Probabilities

Training: Probabilities

- A mapping between a source- and a target word can be represented through a sequence of n segment-pairs

$\langle t_1, s_1 \rangle, \langle t_2, s_2 \rangle \dots \langle t_n, s_n \rangle$

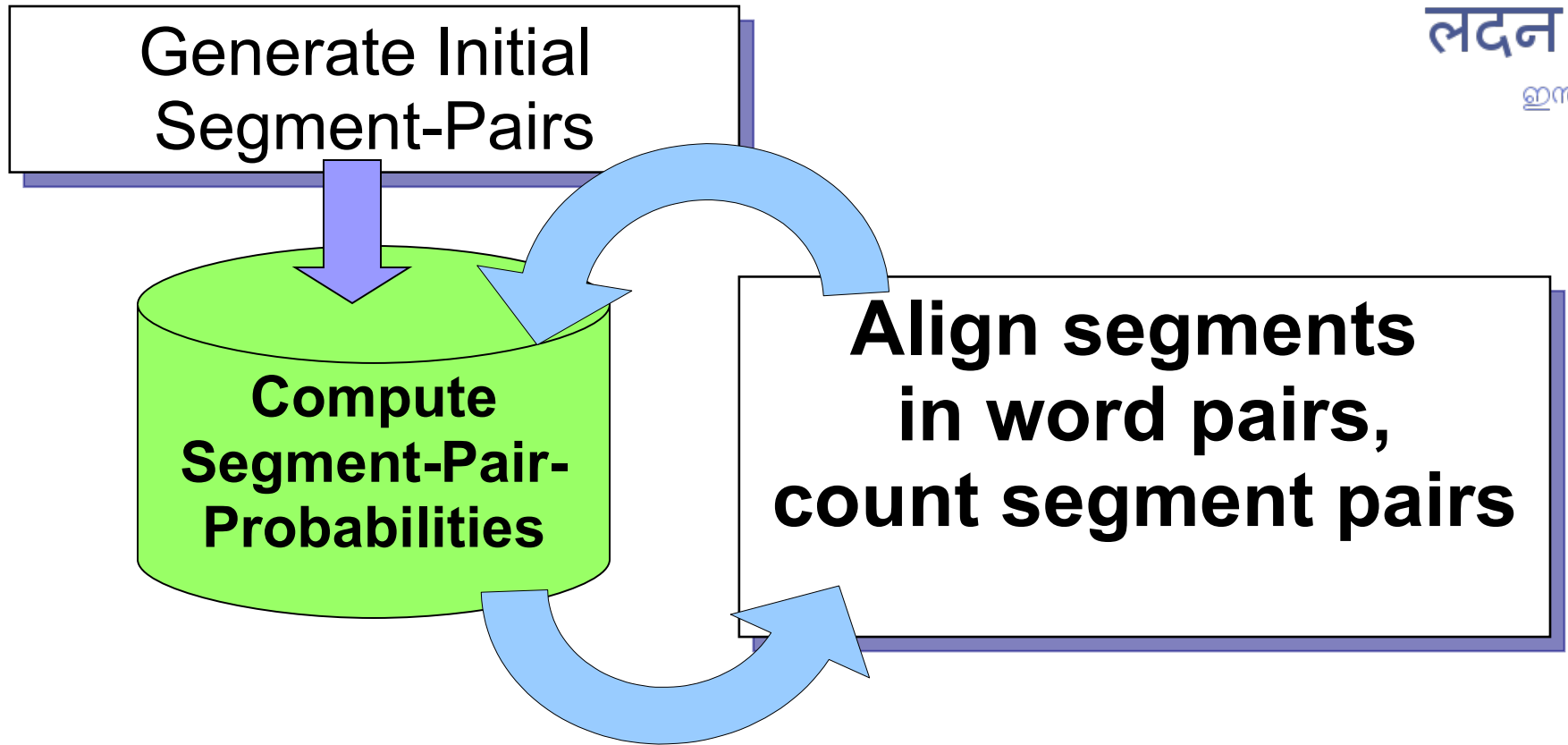
e.g. $\langle न ,na \rangle \langle म , ma \rangle \langle स्त, st \rangle \langle े, e \rangle$

- The probability of this sequence can be calculated as

$$\prod_n P(t_n | s_n) P(t_n | t_{n-1})$$

- $P(t_n | s_n) = \text{count}(\langle t_n, s_n \rangle) / \text{count}(s_n)$
- $P(t_n | t_{n-1})$ is a (character-) bi-gram language model

Iterative Training



Training: Alignment

- Input is Word-pairs:
Source-words and their transliterations
- A Viterbi- type algorithm is used to find the most probable segment-sequence for the source-target-word pair

Align: नमस्ते <> namaste

े							लंदन
त							உரு
ं							
स							
म							
न							
	n	a	m	a	s	t	e

Align: नमस्ते <> namaste

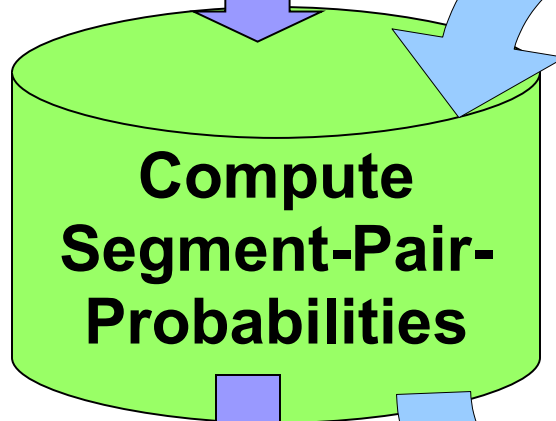
े							P(े e)
त						P(त t)	
ः					P(ः s)		
स		P(स ama)					
म	P(म n)						
न							
	n	a	m	a	s	t	e

Training: Alignment

- This is done for all examples in the training data
- Smoothing guarantees that previously unseen segment pairs are also considered
- Pairs used in the alignments are then counted
- These counts are used to update probabilities

Iterative Training

Generate Initial
Segment-Pairs



Align segments
in word pairs,
count segment pairs

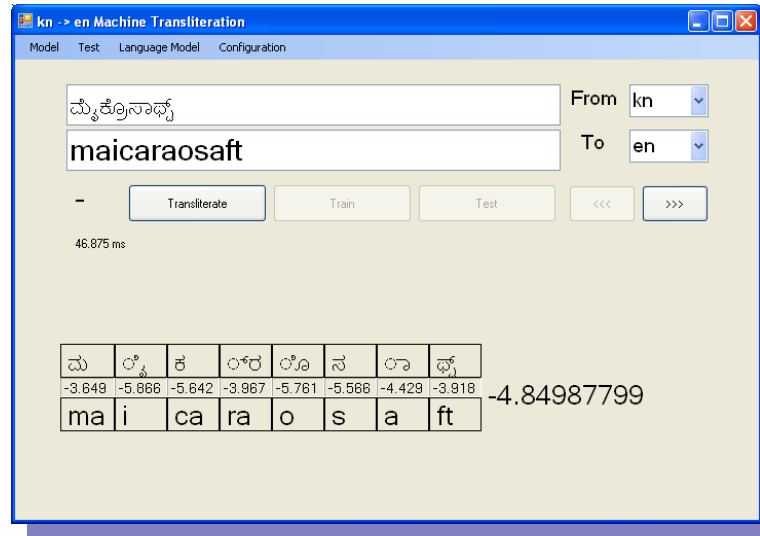
Decodig

Decoding

- Creates the best target-language transliteration for a given source-language word
- Based on the pair-probabilities estimated in training
- Uses a Viterbi-type algorithm

PART III: Results

Part III: Results & Prototype-Demo



Generation



From

To

0

Transliterate

Train

Test

<<<

>>>

status



ಮೈತ್ರೋಸಾಧ್ಯ

From kn

To en

-

Transliterate

Train

Test

<<<

>>>

status

ಮೈಕರಾಸಾಫ್ಟ್

From kn

maicaraosaft

To en

-

Transliterate

Train

Test

<<<

>>>

46.875 ms

ಮ	ೈ	ಕ	ರ	ೊ	ಸ	ಾ	ಫ್ಟ್
-3.649	-5.866	-5.642	-3.967	-5.761	-5.566	-4.429	-3.918
ma	i	ca	ra	o	s	a	ft

-4.84987799

Comparison of candidates



ಮೈತ್ರೋಸಾಧ್ಯ

From kn

To en

-

Transliterate

Train

Test

<<<

>>>

status



ಮೈಕ್ರೋಸಾಫ್ಟ್

From kn

microsoft

To en

-

Transliterate

Train

Test

<<<

>>>

status

ಮೈಕ್ರೊಸಾಫ್ಟ್

From kn

microsoft

To en

-

Transliterate

Train

Test

<<<

>>>

62.5 ms

ಮ	ೈ	ಕ್	ರೊ	ನ	ಾಫ್ಟ್
-6.616	-5.135	-6.102	-4.584	-4.754	-5.736
m	i	c	ro	s	oft

-5.48780517



ಮೈತ್ರೋಸಾಧ್ಯ

From kn

To en

-

Transliterate

Train

Test

<<<

>>>

status



ಮೈಕೊಸಾಫ್ಟ್

From kn

bangalore

To en

-

Transliterate

Train

Test

<<<

>>>

status

From

To

-

Transliterate

Train

Test

<<<

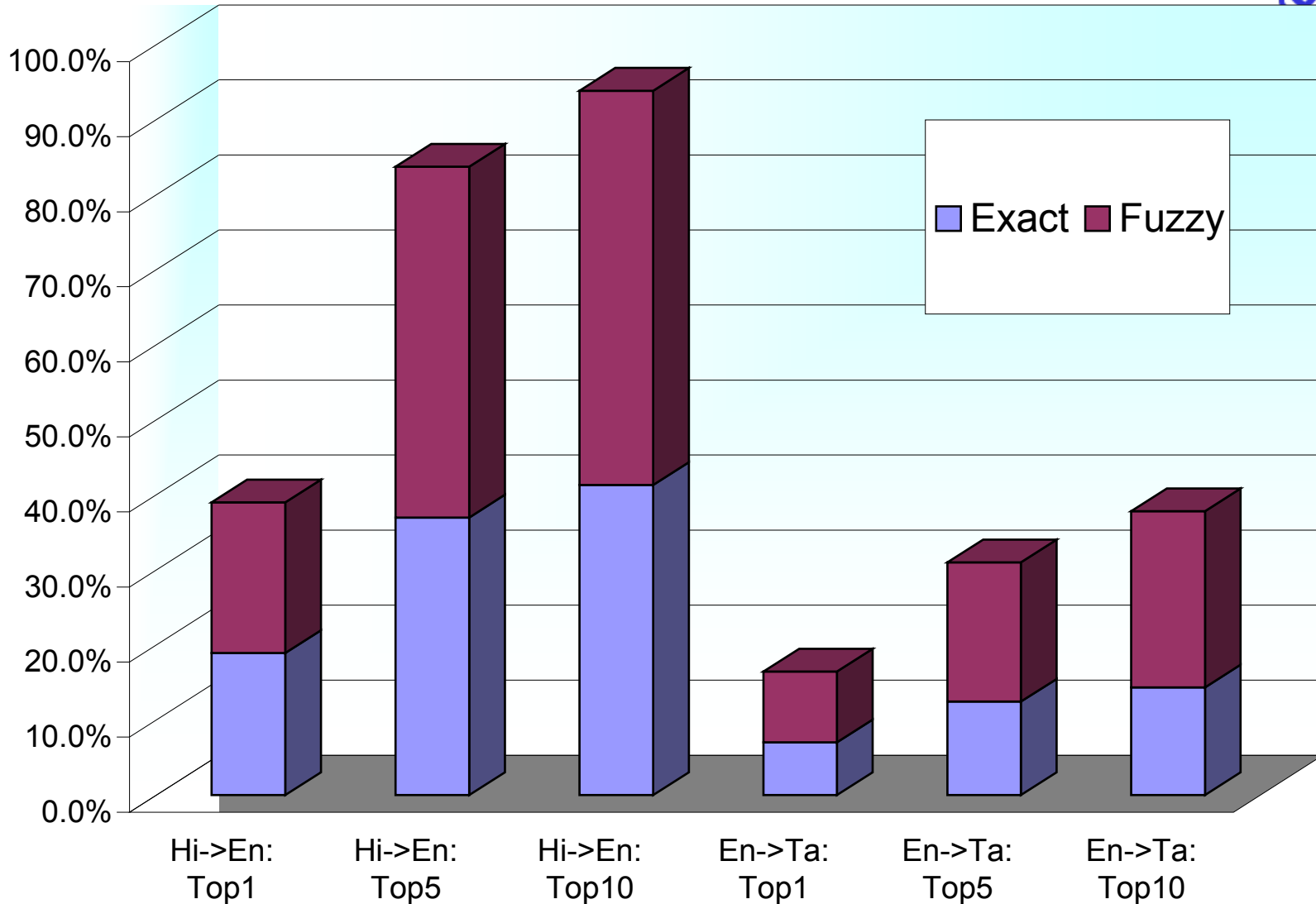
>>>

109.375 ms

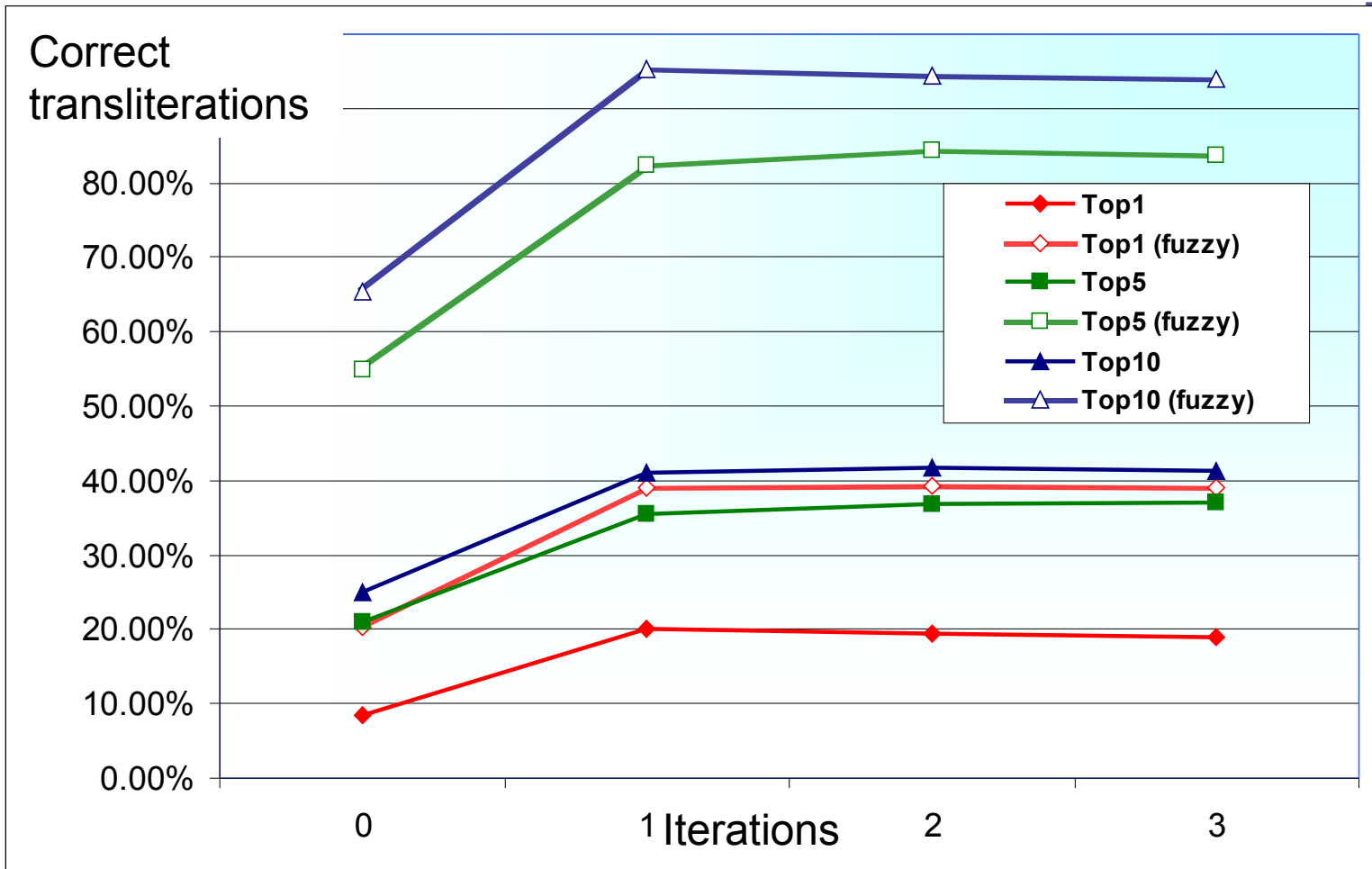
ಮೈಕೆ	಼	ರೊನ	ಾ	ಫ್ಟ್	಼
-15.740	-5.278	-14.560	-3.756	-15.535	-4.082
ba	n	g	a	lor	e

-9.82518544

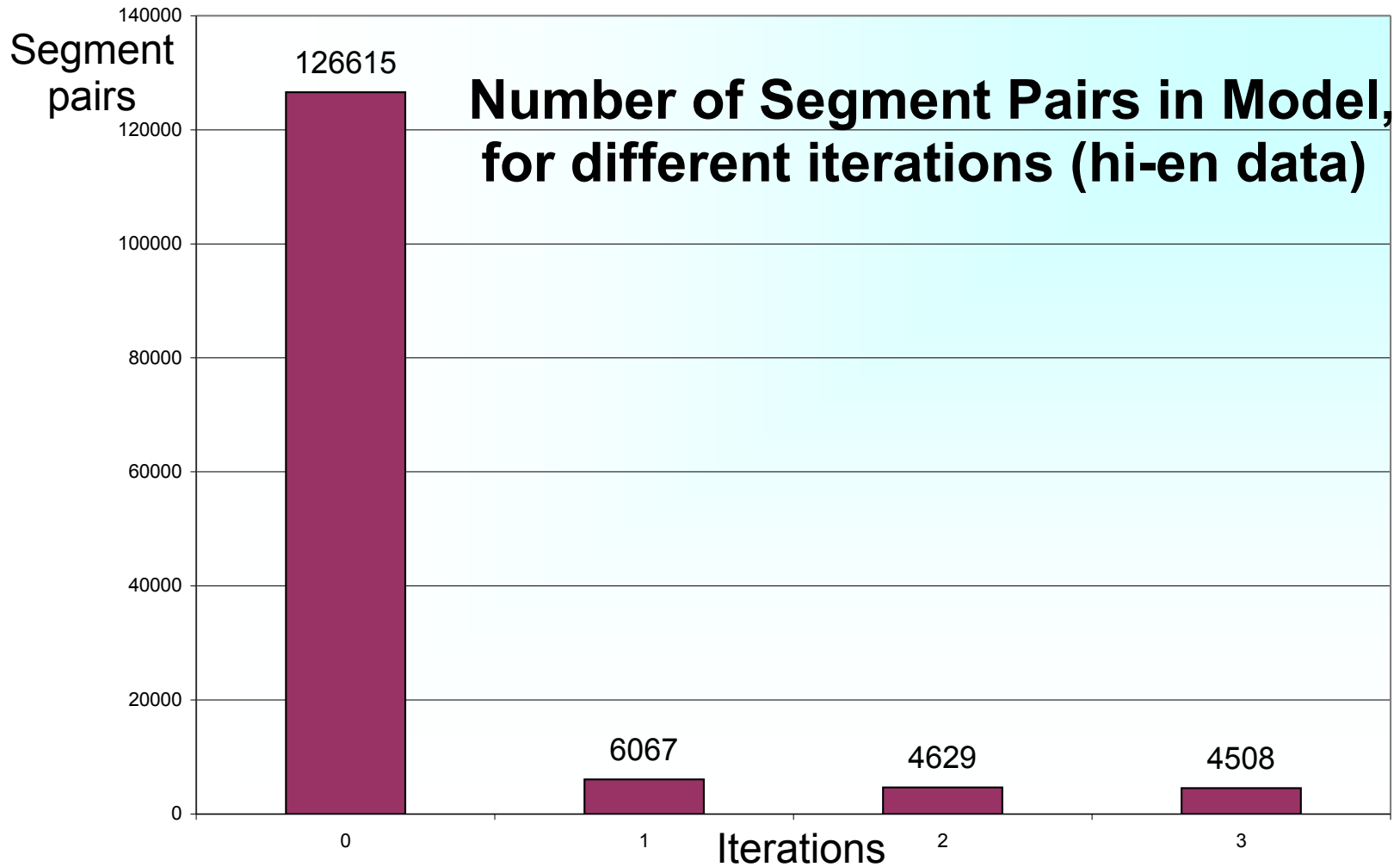
Results (1)



Results (2)



Results (3)



Current trends:

- Transliteration and Named Entity Recognition (NER) in translated text:
 - complementing NER and transliteration to identify cognates in bi-text

