An Introduction to Machine Translation

Anoop Kunchukuttan
Pratik Mehta

under the guidance of
Prof. Pushpak Bhattacharyya

Center for Indian Language Technology
www.cfilt.iitb.ac.in

Department of Computer Science & Engineering
IIT Bombay

anoopk@cse.iitb.ac.in
pratiks@cse.iitb.ac.in
Machine Translation
What is Machine Translation?

Automatic conversion of text/speech from one natural language to another

e.g.

- Be the change you want to see in the world
- वह परिवर्तन बनो जो संसार में देखना चाहते हो
Why do we need machine translation?

- 4 language families
- 22 scheduled languages
- 11 languages with more than 25 million speakers
- 30 languages with more than 1 million speakers
- Only India has 2 languages in the world’s 10 most spoken languages
- 7-8 Indian languages in the top 20 most spoken languages
Translation Usecases

• Government
  - Administrative requirements
  - Education
  - Security

• Enterprise
  - Product manuals
  - Customer support

• Social
  - Travel (signboards, food)
  - Entertainment (books, movies, videos)
Translation under the hood

- Cross-lingual Search
- Cross-lingual Summarization
- Building multilingual dictionaries

Any multilingual NLP system will involve some kind of machine translation at some level
Why study machine translation?

- One of the most challenging problems in Natural Language Processing
- Pushes the boundaries of NLP
- Involves analysis as well as synthesis
- Involves all layers of NLP: morphology, syntax, semantics, pragmatics, discourse
- Theory and techniques in MT are applicable to a wide range of other problems like transliteration, speech recognition and synthesis
Why is machine translation difficult?

Language Divergence: the great diversity among languages of the world

- Word order: SOV (Hindi), SVO (English), VSO, OSV,
- Free (Sanskrit) vs rigid (English) word order
- Analytic (Chinese) vs Polysynthetic (Finnish) languages
- Different ways of expressing same concept
- Case marking systems
- Language registers
- Inflectional systems [infixing (Arabic), fusional (Sanskrit), agglutinative (Marathi)]

... and much more
Why is machine translation difficult?

- Ambiguity
  - Same word, multiple meanings:
  - Same meaning, multiple words: जल, पानी, नीर (water)
- Word Order
  - Underlying deeper syntactic structure
  - Phrase structure grammar?
  - Computationally intensive
- Morphological Richness
  - Identifying basic units of words
Q. Why not just translate all the words in a sentence and put them together?

- "The cold war came to a dramatic end.
  “शीत युद्ध आया एक नाटकीय समाप्ति पर”

- “उसने मेरे जीवन से पर्दा उठाया”
  “He my life from curtain lifted”
  “He lifted a curtain from my life”
  (even after rearranging)
Taxonomy of MT systems

MT approaches

Knowledge-based:
  Rule-based MT

Data Driven:
  Machine Learning based MT

Interlingua based
Transfer based
Example-based MT
Statistical Machine Translation
Rule-Based Machine Translation (RBMT)

• Translation can be seen as a three-step process:
  - Analysis
  - Transfer
  - Generation

• In RBMT, human-created rules are used at every stage of the A-T-G process.

Abstract representation (Interlingua)
Stage 1: Analysis

- Human languages are inherently ambiguous
- Hard to carry over meaning without performing disambiguation
- Example:
  - I deposited ₹1000 in the bank
  - The girl sat dreaming by the river bank.
- In RBMT, ambiguity resolved during analysis stage using hand-crafted rules.
- Resolving all different types of ambiguities is expensive and in certain cases avoidable
Analysis: Types of ambiguity

1. Lexical ambiguity
   - Part of Speech: “We saw her duck.”
   - Sense ambiguity: “She went to the bank to bury all her money.”

2. Boundary ambiguity
   1. Multiwords: “by and large”, “in short”, “every which way”
   2. Chunks:
   3. Phrases: “Spill the beans”, “kick the bucket”
   4. Clauses
Analysis: Types of ambiguity

3. Attachment ambiguity

4. Presuppositional Ambiguity

“Maria solved the problem too.”
Stage 1: Analysis

- “Depending on the depth of analysis, the A-stage in RBMT is a full-fledged NLP enterprise.”
  - Machine Translation: Chapter 5, Pushpak Bhattacharyya

- If the languages are typologically close, detailed disambiguation is not necessary.
Stage 2: Transfer

- Complete disambiguation is unrealistic
- Bilingual mapping cannot cover all lexemes because of new terms, names and coinages
- Therefore, getting to a universal level of abstraction is not possible
Transfer: How to do it?

- We perform analysis up to some level and then perform transfer into the target language using a bilingual dictionary.
- Such a dictionary offers word and phrase mappings from the source language to the target.
Stage 3: Generation

- RBMT arranges for morphology synthesis of lemmas that come from the bilingual dictionary lookup
- We also perform syntax ordering at this stage
- Example to explain
- Generation can be as complex as the analysis stage if native speaker acceptability is to be ensured
Types of RBMT systems

- Interlingua-based
- Transfer-based
Interlingua-based RBMT

• If an MT system aims to ascend to the tip of the Vauquois triangle and descend from there, we have an interlingua based system
What is Interlingua?

- An artificial, man-made language for representing meaning in computers
- It enables representation of meaning without any ambiguity
- Given a piece of text, its interlingual representation will represent the following uniquely:
  - Lexical knowledge
  - Structural knowledge
  - Discourse knowledge
- Thus, any interlingual representation will have:
  1. Represented all words in their disambiguated form
  2. Clearly identified all word groups like multiwords
  3. Resolved all structural ambiguities like attachment
  4. Resolved all discourse ambiguities like co-reference
- Example: UNL
UNL: Universal Networking Language

- UNL is an interlingua proposed by the United Nations university in 1996
- A sentence in UNL is expressed across 3 dimensions:
  - Word knowledge
  - Conceptual knowledge
  - Speech acts (speaker's view, aspect, time of event, etc.)
To convert a natural language sentence to UNL:

- Convert the sentence into a hypergraph with *concepts* as nodes and *relations* as directed arcs
- Concepts are represented as language-independent Universal Words
  - e.g. Liquor $\rightarrow$ drink(icl > liquor)
  - Every UW has restrictions that disambiguate to only one sense of the word.
- Relation labels capture conceptual knowledge.
  - They are expressed as $\text{rel}(\text{UW}_1, \text{UW}_2)$
  - There are 46 relation labels in the UNL today

- Speaker's view, aspect, time of event etc. are captured by attribute labels
On Sunday in Kolkata, Sachin donated to the cricket museum the bat with which he scored his hundredth century at Bangladesh.

Source: Machine Translation by Pushpak Bhattacharyyya
Translation using Interlingua (1/2)

• Analysis
  – POS tagging
  – Parsing
  – Clause marking
  – Text simplification
  – Named Entity recognition (NER)
  – Word Sense Disambiguation (WSD)
Translation using Interlingua (2/2)

• Generation
  – Lexeme selection
  – Case identification
  – Morphology Generation
  – Function word insertion
  – Syntax planning
Transfer-based MT

- Transfer-based MT does not go up to the tip of the Vauquois triangle.
- Instead, we choose an appropriate level of transfer and write transfer-rules for the language-pair.
- The transfer rules guide the transformation of sentence structure across that language-pair.
Transfer Rules

- Used when two languages differ in structure

- Definition:
  - $T: \text{REPs} \rightarrow \text{REPt}$
  - Transfer rule ($T$) maps representation of source language sentences (REPs) to target language sentences (REPt)
  - e.g: $T: \langle V \ NP \rightarrow NP \ V \rangle$
    - This will swap the constituents of a verb phrase from an SVO (Subject-Verb-Object) to SOV (Subject-Object-Verb) word order.
Transfer-based MT: The process

Diagram: Machine Translation, Prof. Pushpak Bhattacharyya
Example based MT

- Example-based MT was founded on the idea that RBMT's approach was not the way humans translate sentences.
- In fact, humans decompose a sentence into fragmental phrases.
- We then translate each of these fragmental phrases and compose them into a single sentence.
- Therefore, it challenged the deep linguistic analysis required by RBMT.
How does Example-based MT work?

• Given a sentence to translate:
  – Match fragments of the sentence to a repository of same-language sentences from a large repository
  – The combination of fragments most similar to the existing examples will be chosen for the transfer stage
  – Alignment is performed by searching for analogs of the phrasal fragments in a parallel body of sentences
  – After the target-language transfer, the fragments are stitched together.
Example-based MT
EBMT: Example

- Input: “He buys a book on international politics”
  - 1. Phrase fragment matching: he buys, a book, international politics.
  - Recombination: vaha antarraaashtriya raajniti par ek kitaab khariidtaa haai
Statistical Machine Translation

Data driven translation
Parallel Corpus

The Noisy Channel Model

<table>
<thead>
<tr>
<th>English</th>
<th>Hindi</th>
</tr>
</thead>
<tbody>
<tr>
<td>So far there is no evidence that there is a limit to the Universe.</td>
<td>ब्रम्हांड की कोई सीमा होने का अब तक कोई सबूत नहीं है।</td>
</tr>
<tr>
<td>The limit is rather on what we can see and how much we can understand.</td>
<td>सीमा बल्कि यही है कि हम क्या देख सकते हैं और हम कितना समझ पাতे हैं।</td>
</tr>
</tbody>
</table>

A very general framework for many NLP problems

Language Model (LM)  
P(e)  
Captures fluency

Translation Model (TM)  
P(f|e)  
Captures fidelity
The SMT Process

Training

- Given: Parallel Corpus
- Output: \( P(e), P(f|e) \)
  - This is model learning
- Learning Objective: Maximize Likelihood
- Offline, one-time process
- Different translation models from different choice of \( P(f|e) \)

\[ P^*(f|e) = \arg \max_e \text{Likelihood}(data; P(f|e)) \]

Decoding

- Given:
  - Sentence \( f \) in language \( F \)
  - \( P(e) \) and \( P(f|e) \)
- Output: Translation \( e \) for \( f \)
- Online process, should be fast
- TM & LM are used for scoring translation candidates

\[ e^* = \arg \max_e P(f|e)P(e) \]
Phrase-based Translation Model

- One of the most successful models
- Widely used in commercial systems like Google Translate
- Basic unit of translation is a phrase
- A *phrase* is just a sequence of words

- Local Reordering
  - Intra-phrase re-ordering can be memorized

<table>
<thead>
<tr>
<th>The Prime Minister of India</th>
<th>भारत के प्रधान मंत्री</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>bharat ke pradhaan maMtrl</td>
</tr>
<tr>
<td></td>
<td>India of Prime Minister</td>
</tr>
</tbody>
</table>

- Sense disambiguation based on local context
  - Neighbouring words help do the right translation

<table>
<thead>
<tr>
<th>heads towards Pune</th>
<th>पुणे की ओर जा रहे है</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>pune ki or jaa rahe hai</td>
</tr>
<tr>
<td></td>
<td>Pune towards go –continuous is</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>heads the committee</th>
<th>समिति की अध्यक्षता करते है</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Samiti kii adhyakshata karte hai</td>
</tr>
<tr>
<td></td>
<td>committee of leading -verbalizer is</td>
</tr>
</tbody>
</table>
So how the model look now?

- Source sentence can be segmented in I phrases
- Then, $p(f|e)$ can be decomposed as:

$$p(f_i^I|e_i^I) = \prod_{i=1}^{I} \phi(f_i|e_i) \cdot d(start_i - end_{i-1} - 1)$$

- $start_i$: start position in $f$ of $i^{th}$ phrase of $e$
- $end_i$: end position in $f$ of $i^{th}$ phrase of $e$

Distortion probability
Phrase Translation Probability
The three As of Machine Translation

- Rule Based MT
  - Analysis
- Statistical MT
  - Alignment
- Example-based MT
  - Analogy
Training a Phrase-based SMT system

• Building the Language Model
• Building the Translation Model
  – Word Alignment (find word-level correspondences)
  – Phrase Extraction (extract phrase pairs)
• Tuning
Building the Language Model

- Probability of a sentence $e$
  - $P(e) = P(e_1,e_2,...,e_k)$
  - $= \prod_{i=1..k} P(e_i|e_{i-1..i-n+1})$
  - Apply Chain Rule of probability
  - Markov Assumption: $i^{th}$ words depends only previous $n-1$ words ($n^{th}$ order Markov model)

- Estimate $P(e_i|e_{i-1..i-n+1})$ from a monolingual corpus
  - e.g. of a bigram (2-gram) language model
  - $P(\text{book}|\text{the}) = \frac{c(\text{the,book})}{c(\text{the})}$
  - A little complication: what happens if $\text{book}$ never comes in the training corpus
  - That's the complicated part of language modelling, let's skip it for now!
Word Alignment

- Central Task in Statistical Machine Translation
- Given a parallel sentence pair, find word level correspondences (*alignment, let's say a*)
But there are multiple possible alignments
But there are multiple possible alignments

How do we find the correct alignment?
Key ideas

• **Co-occurrence of words**
  - Words which occur together in the parallel sentence are likely to be translations 
    (*higher* $P(f|e)$)
  - Alignments which have more likely word-translation pairs are more likely (*higher* $P(a)$)
  - *It's a chicken-and-egg problem!*
  - How to actually find the best alignment?

• **Expectation-Maximization Algorithm**
  - Find the best *hidden* alignment
  - *A key algorithm for various machine learning problems*
    • Start with a random alignment
    • *Find* $P(f|e)$ *given the alignments*
    • Now compute alignment probabilities $P(a)$ with these new translation probabilities
    • Do this repeatedly till $P(f|e)$ does not change
At the end of the process
Learning Phrase Tables from Word Alignments

- Leverages word alignments learnt from IBM models
- Word Alignment: reliable input for phrase table learning
  - high accuracy reported for many language pairs
- Central Idea: A consecutive sequence of aligned words constitutes a "phrase pair"
Extracting Phrase Pairs

<table>
<thead>
<tr>
<th>प्रोफेसर</th>
<th>सी.एन.एर</th>
<th>राव</th>
<th>को</th>
<th>भारतरतन</th>
<th>से</th>
<th>सम्मानित</th>
<th>किया</th>
<th>गया</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prof C.N.R. Rao was honoured with the Bharat Ratna</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Phrase Pairs “consistent” with word alignment

- Consistent
- Inconsistent
- Consistent

Source: SMT, Phillip Koehn
### Examples

<table>
<thead>
<tr>
<th>Professor CNR</th>
<th>प्रोफेसर सी.एन.आर</th>
</tr>
</thead>
<tbody>
<tr>
<td>Professor CNR Rao</td>
<td>प्रोफेसर सी.एन.आर राव</td>
</tr>
<tr>
<td>Professor CNR Rao was</td>
<td>प्रोफेसर सी.एन.आर राव को</td>
</tr>
<tr>
<td>honoured with the Bharat Ratna</td>
<td>भारतरत्न से सम्मानित</td>
</tr>
<tr>
<td>honoured with the Bharat Ratna</td>
<td>भारतरत्न से सम्मानित किया</td>
</tr>
<tr>
<td>honoured with the Bharat Ratna</td>
<td>भारतरत्न से सम्मानित किया गया</td>
</tr>
<tr>
<td>honoured with the Bharat Ratna</td>
<td>को भारतरत्न से सम्मानित किया गया</td>
</tr>
</tbody>
</table>

26 phrase pairs can be extracted from this table.
Computing Phrase Translation Probabilities

• Estimated from the relative frequency:

\[ \phi(f|e) = \frac{\text{count}(e,f)}{\sum_{f_i} \text{count}(e,f_i)} \]

<table>
<thead>
<tr>
<th>Prime Minister of India</th>
<th>भारत के प्रधान मंत्री (India of Prime Minister)</th>
<th>0.75</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prime Minister of India</td>
<td>भारत के भूतपूर्व प्रधान मंत्री (India of former Prime Minister)</td>
<td>0.02</td>
</tr>
<tr>
<td>Prime Minister of India</td>
<td>प्रधान मंत्री (Prime Minister)</td>
<td>0.23</td>
</tr>
</tbody>
</table>
Tuning

• Learning feature weights from data – $\lambda_i$
• Minimum Error Rate Training (MERT)
• Search for weights which minimize the translation error on a held-out set (tuning set)
  – Translation error metric : $(1 - BLEU)$

Source: SMT, Phillip Koehn
Decoding

- Find best translation among a very large number of possible translations
- NP-hard problem: 10-word sentence, 5 translations per word: $10^5 \times 10! \approx 362$ billion possible translations
- Look for approximate solutions
  - Restrict search space: *some word orders are not possible*
  - Incremental construction and scoring
  - Remove candidates that are unlikely to eventually generate good translations
Search Space and Search Organization

- Each hypothesis is scored using the SMT model
- Hypotheses are maintained in a priority queue (called stack decoding historically)
- Limit to the reordering window for efficiency
References

• Introductory textbooks

• Other introductory material
  – Kevin Knight's MT workbook
    www.isi.edu/natural-language/mt/wkbk.pdf
  – ICON 2013 tutorial on Statistical Machine Translation
References (2)

- Getting hands on
  - Moses
    http://www.statmt.org/moses/
  - Google Translate API
    https://cloud.google.com/translate/
  - Indic NLP Library
    https://github.com/anoopkunchukuttan/indic_nlp_library
  - IITB SMT tools for Indian languages
    http://www.cfilt.iitb.ac.in/static/download.html
Pre- and Post- editing

- Machine translation systems are just like people
  - They are unpredictable
  - They are slow to learn
  - **They make mistakes**

- To work around their short-comings, we can either:
  - Make their work easier (pre-editing)
  - Correct their mistakes (post-editing)
Pre-editing: Why do it?

- It is the process of adjusting text before automatically translating it
- Can reduce complexity of search for the MT system
- Helps improve raw MT output
- Reduces the amount of work required for human post-editing
Pre-editing: Processes

1. Morphological segmentation
2. Changing word-order of source language sentences (pre-ordering)
Morphological Segmentation

- Useful for agglutinative languages like Marathi, Tamil and Malayalam:
  e.g. Marathi "घरासमोरच्या" vs the Hindi “घर के सामने”

- The agglutination causes data sparsity

- We simply separate all the morphemes from the root word
  - e.g. “घरासमोरच्या” → घरा समोर च्या
  - Now we can learn all these words and their occurrences separately

- Morfessor is a tool that can learn to do this given any text corpora using unsupervised machine learning

- IndicNLP (http://www.anoopkunchukuttan.github.io/indic_nlp_library ) offers trained morphological analyzers for 12 Indian languages!
Pre-ordering: Why we need it

- Transfer of word-order from source language to target language is hard if the languages differ greatly in word-order.
- Translating from English – an SVO language – is made greatly difficult because of the word order gap.
- In SMT, it is handled by Lexical Reordering- an algorithm that does good at reordering in short windows.
- But words that have to travel very long often don't make it there.
- A very long verb phrase can keep a decoder busy for very long – and yet get poor results.
- So we can shift the word order of the source sentences in advance to make it easier for the MT system.
Preordering: Example

• Rule-based MT requires rules for structural transformation between languages

• Similar rules for transforming word-order for a particular language-pair are required here

• Example: A SVO → SOV conversion for Eng-Hin MT system using hand-written rules gives the following results:
  – “Do not put kajal in the eyes.” → “the eyes in kajal put not Do.”

• The second sentence better matches word-order for the Hindi translation: “आँखों में काजल मत लगाओ”

• The decoder will have to spend lesser time finding a translation candidate with the best probability score.
Post-editing

- The output of an MT system is quite often erroneous
- Post-editing stage is used to identify and correct errors of the best translation candidate
- Words that were not learned during the training phase of SMT cannot be translated to the target language
- Such Out Of Vocabulary (OOV) words can be handled using Transliteration
- Transliteration achieves native speaker readability for the final output
- Example:

<table>
<thead>
<tr>
<th>Source:</th>
<th>Nature has placed this very important control-room in a very safe place at the centre of the head.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample SMT output:</td>
<td>पर्यावृत्ति की इस अत्यंत महत्वपूर्ण control-room सिर के मध्य में एक बहुत ही सुरक्षित स्थान पर रखा गया है</td>
</tr>
<tr>
<td>After transliteration:</td>
<td>पर्यावृत्ति की इस अत्यंत महत्वपूर्ण कंट्रोल-रूम सिर के मध्य में एक बहुत ही सुरक्षित स्थान पर रखा गया है</td>
</tr>
</tbody>
</table>

Transliteration courtesy Brahminet (http://www.cfilt.iitb.ac.in/brahminet/)
Richer Translation Models

- Syntax based SMT
- Factor based SMT
- Whole Document Translation
Some more Interesting Problems

• Translation among Related Languages

• Scaling to larger corpora

• Deep learning and Machine Translation
<table>
<thead>
<tr>
<th>English</th>
<th>Hindi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jagdish Tytler is accused of leading a mob during the 1984 riots.</td>
<td>दिल्ली की एक अदालत ने हकम दिया है कि कांग्रेस नेता और पूर्व मंत्री जगदीश टाइटलर के खिलाफ 1984 सिख विरोधी दंगा मामले में फिर से जांच शुरू की जाए.</td>
</tr>
<tr>
<td>The court has ordered the reopening of a case against this Congress Party leader for his involvement in anti-Sikh riots in 1984.</td>
<td>केंद्रीय जांच एजेंसी सीबीआई की सिफारिश पर दिल्ली की एक कोर्ट ने पहले जगदीश टाइटलर के खिलाफ मामले को बंद करने की इजाजत दी थी.</td>
</tr>
<tr>
<td>Jagdish Tytler was originally cleared by the Central Bureau of Investigation (CBI).</td>
<td>दिल्ली से सांसद रह चुके जगदीश टाइटलर पर आरोप लगाते रहे हैं कि उन्होंने 1984 में लोगों को सिख विरोधी दंगों के दौड़ने का अंदाजा बढ़ाया था.</td>
</tr>
<tr>
<td>The 1984 riots began following the assassination of Mrs Gandhi.</td>
<td>जगदीश टाइटलर कांग्रेस के तीन अहम नेताओं में से एक है जिनके खिलाफ सिख विरोधी दंगों को लेकर आरोप लगाते रहे हैं.</td>
</tr>
</tbody>
</table>

We could go further .... Unsupervised translation
Speech-to-Speech Translation
Image Text to Image Text Translation

Translation on smaller devices
Transliteration
You are in Kerala … waiting to travel by bus

Not a hypothetical situation …. Read this:
How do you translate Xi Jinping?

Xi Jinping is the President of China

शी चिनफिंग चीन के राष्ट्रपति है

Ok, we got lucky here … but there are so many names you will not find in any corpus
Transliteration can simplify Translation

<table>
<thead>
<tr>
<th>Hindi</th>
<th>Punjabi Translation</th>
<th>Hindi-Punjabi Transliteration</th>
</tr>
</thead>
<tbody>
<tr>
<td>यदि श्वास प्रणालिका में सूजन आ जाये तब भी रक्त मूढ़ के रास्ते बाहर आने लगता है।</td>
<td>सेवत माग पूर्णी किस मेंस भा मणे उट ही कुठ पृंथ से उम्म्डे घाट भाँटट सहाना है। जेकर नाह प्रणाली विच सोज आ जाये तद वी खून मूढ़ हे रास्ते बाहर आउण लगदा है।</td>
<td>भाग्य माण पूर्णी में मृत भा मणे उट ही कुठ पृंथ से उम्म्डे घाट भाँटट सहाना है। आदि साह प्रणाली में सूजन आ जाये तद वी रक्त मूढ़ हे रास्ते बाहर आउण लगदा है।</td>
</tr>
</tbody>
</table>
Some Concepts

**Natural Language**: A system of communication among humans with sound

**Script**: A system of symbols for representing language in writing
- *A language can have multiple scripts:*
  - Sanskrit is written in many scripts (Devanagari, Malayalam, Tamil, Telugu, Roman, etc.)
- *A script can be used for multiple languages*
  - Devanagari is used to write Sanskrit, Hindi, Marathi, Konkani, Nepali

**Phoneme**: basic unit of sound in a language that is meaningful

**Grapheme**: basic distinct unit of a script
- *A phoneme can be represented by multiple graphemes*
  - cut, dirt
- *A grapheme can be used to represent multiple sounds*
  - cut, put
Transliteration is the conversion of a given name in the source language (from source script) to a name in the target language (target script), such that the target language name is:

- phonemically equivalent to the source name
  
  मुंबई → Mumbai

- conforms to the phonology of the target language
  
  ਨਰੰਦਰ → ਨਰੰਦਰ (ਨਰੰਦਰ)

- matches the user intuition of the equivalent of the source language name in the target language, considering the culture and orthographic character usage in the target language
  
  ആലപ്പുഴ (aalappuzha) → Alappuzha
Isn't it easy to just map characters from one script to another?

- **Local spelling conventions**
  
  लता in Roman: Latha (South India) vs Lata (North India)
  
  Laxmi → लक्ष्मी

- **Missing sounds**
  
  കൊഴിക്കോട് (kozhikkoT) → 科什ikkod (koShikkod)

- **Transliterate or translate**
  
  കൊഴിക്കോട് (kozhikkoT) → Calicut

- **Transliteration variants**
  
  मुंबई, मुम्बई
Why English spellings caused trouble in school ...

Ambiguity in character to sound mapping

**ionize vs nation**

*fish* can be pronounced as *ghoti*

- gh as in *tough*
- o as in *women*
- ti as in *nation*
... and Hindi spellings didn't

Unambiguous mapping from character to sound

Rememember the **varnamala**? – organized according to scientific principles

<table>
<thead>
<tr>
<th>Voicing →</th>
<th>aghoṣa (Plosive)</th>
<th>ghoṣa (Nasal)</th>
<th>antastha (Approximant)</th>
<th>ṛṣma/samghaṣhri (Fricative)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>allophonic</td>
<td></td>
<td>allophonic</td>
<td></td>
</tr>
<tr>
<td>Aspiration →</td>
<td>alpaprāṇa</td>
<td>mahāaprāṇa</td>
<td>alpaprāṇa</td>
<td>mahāaprāṇa</td>
</tr>
<tr>
<td>kanyha (Guttural)</td>
<td>क</td>
<td>ख</td>
<td>ग</td>
<td>घ</td>
</tr>
<tr>
<td>tālavya (Palatal)</td>
<td>च</td>
<td>छ</td>
<td>ज</td>
<td>झ</td>
</tr>
<tr>
<td>mūrdhanya (Retroflex)</td>
<td>ट</td>
<td>ठ</td>
<td>ड</td>
<td>ढ</td>
</tr>
<tr>
<td>dantya (Dental)</td>
<td>त</td>
<td>थ</td>
<td>द</td>
<td>ध</td>
</tr>
<tr>
<td>oṣṭhya (Labial)</td>
<td>प</td>
<td>फ</td>
<td>ब</td>
<td>भ</td>
</tr>
</tbody>
</table>
The extent of Devanagari-like scripts
How do we solve the transliteration problem?

- Transliteration is very similar to translation
- Instead of words, we have characters
- However, it is much simpler
  - No reordering
  - Small vocabulary (except Chinese and Japanese Kanji)
  - Regular grammar
- Similar to Vouquois triangle, you can transliterate at different levels:
  - Phoneme (like transfer based MT)
  - Grapheme (like direct MT)
References

• About Scripts
  – Omniglot: http://www.omniglot.com/
  – Wikipedia pages on Devanagari & Brahmi script

• About Transliteration
  – Karimi, Sarvnaz, Falk Scholer, and Andrew Turpin. "Machine transliteration

• Hands on
  – Google Transliterates
    http://www.google.com/inputtools/
  – Brahmi-Net: IITB's transliteration system
    http://www.cfilt.iitb.ac.in/brahminet/
Thank You!
Acknowledgments

• Some of slides & images have been borrowed from
  - *Machine Translation*, Pushpak Bhattacharyya
  - Wikipedia