An Introduction to Machine Translation and Transliteration

Anoop Kunchukuttan

Center for Indian Language Technology, Indian Institute of Technology Bombay

anoopk@cse.iitb.ac.in

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

THINK Summer School on Machine Learning. Vidyalankar Institute of Technology, Mumbai 19th June 2017
Outline

- Introduction
- Machine Translation Paradigms
- Phrase-based SMT
- Extensions to Phrase-based SMT
- Evaluation of Machine Translation
- Neural Machine Translation
- Summary
Outline

- Introduction
- Machine Translation Paradigms
- Phrase-based SMT
- Extensions to Phrase-based SMT
- Evaluation of Machine Translation
- Neural Machine Translation
- Summary
What is Machine Translation?

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

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?

- 5+1 language families
  - Indo-Aryan (74% population)
  - Dravidian (24%)
  - Austro-Asiatic (1.2%)
  - Tibeto-Burman (0.6%)
  - Andaman languages (2 families?)
  - + English (West-Germanic)

- 22 scheduled languages

- 11 languages with more than 25 million speakers
  - 30 languages with more than 1 million speakers
  - Only India has 2 languages (+English) in the world’s 10 most spoken languages
  - 7-8 Indian languages in the top 20 most spoken languages
Machine 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 should you 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
Outline

● Introduction
● **Machine Translation Paradigms**
● Phrase-based SMT
● Extensions to Phrase-based SMT
● Evaluation of Machine Translation
● Neural Machine Translation
● Summary
**Approaches to build MT systems**

Knowledge based, Rule-based MT

- Transfer-based
- Interlingua based

Data-driven, Machine Learning based MT

- Example-based
- Statistical
- Neural
Rule-based MT

- Rules are written by **linguistic experts** to analyze the source, generate an intermediate representation, and generate the target sentence.
- Depending on the depth of analysis: interlingua or transfer-based MT.

**Interlingua based MT**

- **Deep analysis, complete disambiguation and language independent representation**

**Transfer based MT**

- **Partial analysis, partial disambiguation and a bridge intermediate representation**
Vauquois Triangle

Translation approaches can be classified by the depth of linguistic analysis they perform.
Problems with rule based MT

- Required linguistic expertise to develop systems
- Maintenance of system is difficult
- Difficult to handle ambiguity
- Scaling to a large number of language pairs is not easy
Example-based MT

**Translation by analogy** ⇒ match parts of sentences to known translations and then combine

**Input:** He buys a book on international politics

1. **Phrase fragment matching:** *(data-driven)*
   
   *he buys*
   *a book*
   *international politics*

2. **Translation of segments:** *(data-driven)*
   
   वह खरीदता है
   एक किताब
   अंतर राष्ट्रीय राजनीति

3. **Recombination:** *(human crafted rules/templates)*
   
   वह अंतर राष्ट्रीय राजनीति पर एक किताब खरीदता है

- Partly rule-based, partly data-driven.
- Good methods for matching and large corpora did not exist when proposed.
Outline

- Introduction
- Machine Translation Paradigms
  - Phrase-based SMT
- Extensions to Phrase-based SMT
- Evaluation of Machine Translation
- Neural Machine Translation
- Summary
Statistical Machine Translation

Phrase based SMT
Let’s begin with a simplified view of Statistical Machine Translation (SMT)!!

### Parallel Corpus

<table>
<thead>
<tr>
<th>English</th>
<th>Hindi</th>
</tr>
</thead>
<tbody>
<tr>
<td>A boy is sitting in the kitchen</td>
<td>एक लडका रसोई में बैठा है</td>
</tr>
<tr>
<td>A boy is playing tennis</td>
<td>एक लडका टेनिस खेल रहा है</td>
</tr>
<tr>
<td>A boy is sitting on a round table</td>
<td>एक लडका एक गोल मेज पर बैठा है</td>
</tr>
<tr>
<td>Some men are watching tennis</td>
<td>कुछ आदमी टेनिस देख रहे है</td>
</tr>
<tr>
<td>A girl is holding a black book</td>
<td>एक लडकी ने एक काली किताब पकडी है</td>
</tr>
<tr>
<td>Two men are watching a movie</td>
<td>दो आदमी चलचित्र देख रहे है</td>
</tr>
<tr>
<td>A woman is reading a book</td>
<td>एक औरत एक किताब पढ़ रही है</td>
</tr>
<tr>
<td>A woman is sitting in a red car</td>
<td>एक औरत एक लाल कार में बैठी है</td>
</tr>
</tbody>
</table>

Machine Learning
**Parallel Corpus**

<table>
<thead>
<tr>
<th>English</th>
<th>Hindi</th>
</tr>
</thead>
<tbody>
<tr>
<td>A boy is <em>sitting</em> in the kitchen</td>
<td>एक लडका रसोई में बैठा है</td>
</tr>
<tr>
<td>A boy is playing <em>tennis</em></td>
<td>एक लडका टेनिस खेल रहा है</td>
</tr>
<tr>
<td>A boy is <em>sitting</em> on a round table</td>
<td>एक लडका एक गोल मेज पर बैठा है</td>
</tr>
<tr>
<td>Some men <em>are watching</em> tennis</td>
<td>कुछ आदमी टेनिस देख रहे है</td>
</tr>
<tr>
<td>A girl is holding a black book</td>
<td>एक लडकी ने एक काली किताब पकडी है</td>
</tr>
<tr>
<td>Two men <em>are watching</em> a movie</td>
<td>दो आदमी चलचित्र देख रहे है</td>
</tr>
<tr>
<td>A woman is reading a book</td>
<td>एक औरत एक किताब पढ़ रही है</td>
</tr>
<tr>
<td>A woman is <em>sitting</em> in a red car</td>
<td>एक औरत एक काले कार में बैठा है</td>
</tr>
</tbody>
</table>

**Machine Learning**

*Learn word/phrase alignments*

*Let’s begin with a simplified view of Statistical Machine Translation (SMT)!!*
<table>
<thead>
<tr>
<th>English</th>
<th>Hindi</th>
</tr>
</thead>
<tbody>
<tr>
<td>A boy is <strong>sitting</strong> in the kitchen</td>
<td>एक लडका रसोई में बैठा है</td>
</tr>
<tr>
<td>A boy is playing <strong>tennis</strong></td>
<td>एक लडका टेनिस खेल रहा है</td>
</tr>
<tr>
<td>A boy is <strong>sitting</strong> on a round table</td>
<td>एक लडका एक गोल मेज पर बैठा है</td>
</tr>
<tr>
<td>Some men <strong>are watching</strong> tennis</td>
<td>कुछ आदमी टेनिस देख रहे है</td>
</tr>
<tr>
<td>A girl is holding a black book</td>
<td>एक लडकी ने एक काली किताब पकडी है</td>
</tr>
<tr>
<td>Two men <strong>are watching</strong> a movie</td>
<td>दो आदमी चलचित्र देख रहे है</td>
</tr>
<tr>
<td>A woman is reading a book</td>
<td>एक औरत एक किताब पढ़ रही है</td>
</tr>
<tr>
<td>A woman is <strong>sitting</strong> in a red car</td>
<td>एक औरत एक काले कार में बैठा है</td>
</tr>
</tbody>
</table>

**Let’s begin with a simplified view of Statistical Machine Translation (SMT)!!**
Let’s formalize the translation process

We will model translation using a **probabilistic model**. Why?
- We would like to have a measure of confidence for the translations we learn
- We would like to model uncertainty in translation

\[
\bar{e} = \arg \max_e P(e|f)
\]

**Model**: a simplified and idealized understanding of a physical process

We must first explain the process of translation
We explain translation using the **Noisy Channel Model**

A very general framework for many NLP problems

Generate target sentence

Channel corrupts the target

Source sentence is a corruption of the target sentence

**Translation** is the process of recovering the original signal given the corrupted signal

\[ P(e|f) = P(e) \times P(f|e) \]

Why use this counter-intuitive way of explaining translation?

- Makes it easier to mathematically represent translation and learn probabilities
- **Fidelity** and **Fluency** can be modelled separately
The SMT Workflow

Training
- Given: Parallel Corpus
- Learn Model: $P(e)$ and $P(f|e)$
- Offline, one-time process
- Learning Objective: Maximize Likelihood

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

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

\[ \bar{e} = \arg \max_{e} P(e) \times P(f|e) \]

Let’s see how to learn translation model $P(f|e)$ and language model $P(e)$
Phrase-based Translation Model

- Let’s see one of the most successful translation models: PBSMT
- Widely used in commercial systems like Google Translate (till recently)
- 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 हैं</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 हैं</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Samiti kii adhyakshanta 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:

$$p(f_{1}^{l}|e_{1}^{l}) = \prod_{i=1}^{I} \phi(f_{i}|e_{i}) \ d(\text{start}_{i} - \text{end}_{i-1} - 1)$$
**Language Model**

*Measures how likely a sentence is* \( P(e) \) *⇒ a proxy for grammatical correctness/fluency*

\[
P(e) = P(e_1,e_2,...,e_k) = \prod_{i=1}^{k} P(e_i|e_{i-1}..e_1) = \prod_{i=1}^{k} P(e_i|e_{i-1}..e_{i-n+1})
\]

**Chain Rule**

**Markov assumption**

*How to estimate* \( P(e_i|e_{i-1}..e_{i-n+1}) \)?

- We can estimate the probabilities by counting from a monolingual corpus
- \( P(\text{book}|\text{the}) = \frac{\#(\text{the,book})}{\#(\text{the})} \)
- A little complication: what happens if *book* never comes in the training corpus
- That's the complicated part of language modelling, let's skip it for now!
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 the Model
Word Alignment

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

Sentence 1
But there are multiple possible alignments

Sentence 2

How do we find the correct alignment?
Key idea

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

Sentence 2
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>C.N.R. Rao</th>
<th>was honoured with</th>
<th>the</th>
<th>Bharat Ratna</th>
</tr>
</thead>
<tbody>
<tr>
<td>सी.एन.आर</td>
<td>राओ</td>
<td>को</td>
<td>भारतरत्न</td>
<td>से</td>
</tr>
<tr>
<td>राव</td>
<td>को</td>
<td>भारतरत्न</td>
<td>से</td>
<td>सम्मानित</td>
</tr>
<tr>
<td>को</td>
<td>भारतरत्न</td>
<td>से</td>
<td>सम्मानित</td>
<td>किया</td>
</tr>
</tbody>
</table>
Phrase Pairs “consistent” with word alignment

consistent

inconsistent

consistent

Source: SMT, Phillip Koehn
# Examples

<table>
<thead>
<tr>
<th>English</th>
<th>Hindi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Professor CNR</td>
<td>प्रोफेसर सी.एन.आर</td>
</tr>
<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>Phrase</th>
<th>Translation in Hindi</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prime Minister of India</td>
<td>भारत के प्रधान मंत्री</td>
<td>0.75</td>
</tr>
<tr>
<td></td>
<td>India of Prime Minister</td>
<td></td>
</tr>
<tr>
<td>Prime Minister of India</td>
<td>भारत के भूतपूर्व प्रधान मंत्री</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>India of former Prime Minister</td>
<td></td>
</tr>
<tr>
<td>Prime Minister of India</td>
<td>प्रधान मंत्री</td>
<td>0.23</td>
</tr>
<tr>
<td></td>
<td>Prime Minister</td>
<td></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
Overall Training Process for PB-SMT

- Parallel training corpus
  - Word aligner (E.g. GIZA++)
  - Phrase pair extraction
  - Phrase tables
  - Language model learner (E.g. SRI, IRST)
  - Language Model
  - Parameter weights
  - Decoder

- Word alignments
  - Distortion model learning
  - Distortion Model
  - MERT Tuning

- Parallel tuning corpus
  - Other Feature Extractors
  - Feature values

18-Dec-2013
SMT Tutorial, ICON-2013 194
Decoding

We have learnt a translation model, how do we translate a new sentence?

- Isn’t it ok to just score all possible translations using the model? 
  $\tilde{e} = \arg \max_e P(e) \times P(f|e)$

- NP-hard problem: 10-word sentence, 5 translations per word: $10^5 \times 10! \approx 362$ billion possible translations $\Rightarrow$ Not possible to score each candidate

- 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
Outline

- Introduction
- Machine Translation Paradigms
- Phrase-based SMT
- Extensions to Phrase-based SMT
- Evaluation of Machine Translation
- Neural Machine Translation
- Summary
We have looked at a basic phrase-based SMT system

This system can learn word and phrase translations from parallel corpora

But many important linguistic phenomena need to be handled

- Rich morphology
- Out of Vocabulary words
- Divergent Word Order
**Language is very productive, you can combine words to generate new words**

Inflectional forms of the Marathi word घर

<table>
<thead>
<tr>
<th>घर</th>
<th>house</th>
</tr>
</thead>
<tbody>
<tr>
<td>घरात</td>
<td>in the house</td>
</tr>
<tr>
<td>घरावरती</td>
<td>on the house</td>
</tr>
<tr>
<td>घराखाली</td>
<td>below the house</td>
</tr>
<tr>
<td>घरामधे</td>
<td>in the house</td>
</tr>
<tr>
<td>घरामागे</td>
<td>behind the house</td>
</tr>
<tr>
<td>घराचा</td>
<td>of the house</td>
</tr>
<tr>
<td>घरामागचा</td>
<td>that which is behind the house</td>
</tr>
<tr>
<td>घरासमोर</td>
<td>in front of the house</td>
</tr>
<tr>
<td>घरासमोरचा</td>
<td>that which is in front of the house</td>
</tr>
<tr>
<td>घरांसमोर</td>
<td>in front of the houses</td>
</tr>
</tbody>
</table>

Hindi words with the suffix वाद

<table>
<thead>
<tr>
<th>वाद</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>साम्यवाद</td>
<td>communism</td>
</tr>
<tr>
<td>समाजवाद</td>
<td>socialism</td>
</tr>
<tr>
<td>पूंजीवाद</td>
<td>capitalism</td>
</tr>
<tr>
<td>जातीवाद</td>
<td>casteism</td>
</tr>
<tr>
<td>साम्राज्यवाद</td>
<td>imperialism</td>
</tr>
</tbody>
</table>

The corpus should contains all variants to learn translations

*This is infeasible!*
Inflectional forms of the Marathi word घर

<table>
<thead>
<tr>
<th>घर</th>
<th>घरात</th>
<th>घरावरती</th>
<th>घराखाली</th>
<th>घरामध्ये</th>
<th>घरामागे</th>
<th>घराचा</th>
<th>घरामागचा</th>
<th>घरासमोर</th>
<th>घरासमोरचा</th>
<th>घरांसमोर</th>
</tr>
</thead>
<tbody>
<tr>
<td>घर</td>
<td>घरात</td>
<td>घरावरती</td>
<td>घराखाली</td>
<td>घरामध्ये</td>
<td>घरामागे</td>
<td>घराचा</td>
<td>घरामागचा</td>
<td>घरासमोर</td>
<td>घरासमोरचा</td>
<td>घरांसमोर</td>
</tr>
</tbody>
</table>

| house | in the house | on the house | below the house | in the house | behind the house | of the house | that which is behind the house | in front of the house | that which is in front of the house | in front of the houses |

Hindi words with the suffix वाद

<table>
<thead>
<tr>
<th>साध्य वाद</th>
<th>समाज वाद</th>
<th>पूरी वाद</th>
<th>जाती वाद</th>
<th>सामाजिक वाद</th>
</tr>
</thead>
<tbody>
<tr>
<td>communism</td>
<td>socialism</td>
<td>capitalism</td>
<td>casteism</td>
<td>imperialism</td>
</tr>
</tbody>
</table>

Break the words into its component morphemes

Now we need to only learn translations for the morphemes

Far more likely to find morphemes in the corpus

Tools for obtaining morphemes: Morfessor and Indic NLP Library
Some words not seen during train will be seen at test time

These are out-of-vocabulary (OOV) words

Names are one of the most important category of OOVs
⇒ There will always be names not seen during training

How do we translate names like Sachin Tendulkar to Hindi?

Note: We want to do a mapping of characters so that they sound the same in Hindi
⇒ We call this process ‘transliteration’. More on transliteration later ...
So far we have seen how to learnt how to translate words and phrases

Let's see how we can generate the correct word order
Getting word order right

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 → VBD NP PP ⇒ VP → PP NP VBD

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

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

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 बॉक्सऑफिस पर 1500 करोड रुपए कमाए

These rules can be written manually or learnt from parse trees
Better methods exist for generating the correct word order

Incorporate learning of reordering is built into the SMT system

Hierarchical PBSMT ⇒ Provision in the phrase table for limited & simple reordering rules

Syntax-based SMT ⇒ Another SMT paradigm, where the system learns mappings of “treelets” instead of mappings of phrases
Outline

● Introduction
● Machine Translation Paradigms
● Phrase-based SMT
● Extensions to Phrase-based SMT
● Evaluation of Machine Translation
● Neural Machine Translation
● Summary
Evaluation of Machine Translation

How do we judge a good translation? Can a machine do this? Why should a machine do this? Because humans take time!
• Assign scores to specific qualities of output
  – Intelligibility: How good the output is as a well-formed target language entity
  – Accuracy: How good the output is in terms of preserving content of the source text

**For example,** I am attending a lecture

मैं एक व्याख्यान बैठा हूँ
*Main ek vyakhyan baitha hoon*

*I a lecture* sit *(Present-first person)*

*I sit a lecture*: Accurate but not intelligible

मैं व्याख्यान हूँ
*Main vyakhyan hoon*

*I lecture* am

*I am lecture*: Intelligible but not accurate.
How is translation performance measured?

The closer a machine translation is to a professional human translation, the better it is.

- A corpus of good quality human reference translations
- A numerical “translation closeness” metric
Preliminaries

• **Candidate Translation(s):** Translation returned by an MT system

• **Reference Translation(s):** ‘Perfect’ translation by humans
Human Evaluation

<table>
<thead>
<tr>
<th>Adequacy</th>
<th>Fluency</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>all meaning</td>
</tr>
<tr>
<td>4</td>
<td>flawless English</td>
</tr>
<tr>
<td>3</td>
<td>most meaning</td>
</tr>
<tr>
<td>4</td>
<td>good English</td>
</tr>
<tr>
<td>3</td>
<td>much meaning</td>
</tr>
<tr>
<td>3</td>
<td>non-native English</td>
</tr>
<tr>
<td>2</td>
<td>little meaning</td>
</tr>
<tr>
<td>2</td>
<td>disfluent English</td>
</tr>
<tr>
<td>1</td>
<td>none</td>
</tr>
<tr>
<td>1</td>
<td>incomprehensible</td>
</tr>
</tbody>
</table>

- Adequacy: Does the output convey the same meaning as the input sentence? Is part of the message lost, added, or distorted?
- Fluency: Is the output fluent? This involves both grammatical correctness and idiomatic word choices.
The most popular metric for MT evaluation: BLEU

Bilingual Language Evaluation Understudy

- Simple metric which computes precision and some notion of recall
- Language Independent
Formulating BLEU (Step 1): Precision

I had lunch now.

Reference 1: मैने अभी खाना खाया
maine abhi khana khaya
I now food ate
I ate food now.

Reference 2: मैने अभी भोजन किया
maine abhi bhojan kiya
I now meal did
I did meal now.

Candidate 1: मैने अब खाना खाया
maine ab khana khaya
I now food ate
I ate food now

Candidate 2: मैने अभी लंच एट
maine abhi lunch ate.
I now lunch ate
I ate lunch(OOV) now(OOV)

matching unigrams: 3, matching bigrams: 1

Candidate 1: 3/4 = 0.75, Candidate 2: 2/4 = 0.5
Similarly, bigram precision: Candidate 1: 0.33, Candidate 2 = 0.33
Precision: Not good enough

Reference: मुझ पर तेरा सुरूर छाया
mujh-par tera suoor chhaaya
me-on your spell cast
Your spell was cast on me

Candidate 1: मेरे तेरा सुरूर छाया
mere tera suoor chhaaya
my your spell cast
Your spell cast my

Candidate 2: तेरा तेरा तेरा सुरूर
tera tera tera suoor
your your your spell

matching unigrams: 4

Unigram precision: Candidate 1: 3/4 = 0.75, Candidate 2: 4/4 = 1
Formulating BLEU (Step 2): Modified Precision

• Clip the total count of each candidate word with its maximum reference count

• \( \text{Count}_{\text{clip}}(n\text{-gram}) = \min(\text{count}, \text{max}_\text{ref}_\text{count}) \)

Reference: मुझ पर तेरा सुरूः छहया
            mujh-par tera suroor chhaaya
            me-on your spell cast
            Your spell was cast on me

Candidate 2: तेरा तेरा तेरा सुरूः
            tera tera tera suroor
            your your your spell

• matching unigrams:
  (तेरा : \( \min(3, 1) = 1 \)) (सुरूः : \( \min(1, 1) = 1 \))

Modified unigram precision: \( \frac{2}{4} = 0.5 \)
Modified n-gram precision

For entire test corpus, for a given n,

\[
p_n = \frac{\sum_{C \in \{\text{Candidates}\}} \text{Count}_{\text{clip}}(\text{n-gram})}{\sum_{C' \in \{\text{Candidates}\}} \text{Count}(\text{n-gram}')}
\]

Modified precision for n-grams

Overall candidates of test corpus

n-gram: Matching n-grams in C

n-gram’: All n-grams in C

This metric prefers shorter candidates translations, hence a brevity penalty is added.
**BLEU score**

Recall -> Brevity Penalty

\[ BP = \begin{cases} 
1 & \text{if } c > r \\
\frac{1}{e^{(1-r/c)}} & \text{if } c \leq r 
\end{cases} \]

Precision -> Modified n-gram precision

\[
p_n = \frac{\sum_{n \text{-gram} \in C} \sum_{C \in \text{Candidates}} \text{Count}_{clip}(n\text{-gram})}{\sum_{C' \in \text{Candidates}} \sum_{n\text{-gram}' \in C'} \text{Count}(n\text{-gram}')} \]

\[
\text{BLEU} = BP \cdot \exp \left( \sum_{n=1}^{N} w_n \log p_n \right)
\]
Why does BLEU score work at all?

Well co-related with human judgments
This is the principle for evaluation of evaluation metrics!
Outline

- Introduction
- Machine Translation Paradigms
- Phrase-based SMT
- Extensions to Phrase-based SMT
- Evaluation of Machine Translation
  - Neural Machine Translation
- Summary
Neural Machine Translation
SMT, Rule-based MT and Example based MT manipulate symbolic representations of knowledge.

Every word has an atomic representation, which can’t be further analyzed.

<table>
<thead>
<tr>
<th>Word</th>
<th>Symbolic Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>home</td>
<td>0</td>
</tr>
<tr>
<td>water</td>
<td>1</td>
</tr>
<tr>
<td>house</td>
<td>2</td>
</tr>
<tr>
<td>tap</td>
<td>3</td>
</tr>
</tbody>
</table>

No notion of similarity or relationship between words:
- Even if we know the translation of home, we can’t translate house if it is an OOV.

Difficult to represent new concepts:
- We cannot say nothing about ‘mansion’ if it comes up at test time.
- Creates problems language model as well ⇒ whole area of smoothing exists to overcome this problem.

Symbolic representations are discrete representations:
- Generally computationally expensive to work with discrete representations.
- e.g. Reordering requires evaluation of an exponential number of candidates.
Neural Network techniques work with distributed representations

- No element of the vector represents a particular word
- The word can be understood with all vector elements
- Hence distributed representation
- But less interpretable

Can define similarity between words
- Vector similarity measures like cosine similarity
- Since representations of home and house, we may be able to translate house

Every word is represented by a vector of numbers

<table>
<thead>
<tr>
<th>Word</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>home</td>
<td>0.5</td>
<td>0.6</td>
<td>0.7</td>
</tr>
<tr>
<td>water</td>
<td>0.2</td>
<td>0.9</td>
<td>0.3</td>
</tr>
<tr>
<td>house</td>
<td>0.55</td>
<td>0.58</td>
<td>0.77</td>
</tr>
<tr>
<td>tap</td>
<td>0.24</td>
<td>0.6</td>
<td>0.4</td>
</tr>
</tbody>
</table>

New concepts can be represented using a vector with different values

Symbolic representations are continuous representations
- Generally computationally more efficient to work with continuous values
- Especially optimization problems
Encode - Decode Paradigm

**Input**

- **Embed**
  - **Embedding**
  - **Source Representation**
  - **Output**

**Encoder**

**Decoder**

- **Entire input sequence is processed before generation starts**
  ⇒ In PBSMT, generation was piecewise

**The input is a sequence of words, processed one at a time**

- While processing a word, the network needs to know what it has seen so far in the sequence
- Meaning, know the history of the sequence processing
- Needs a special kind of neural: Recurrent neural network unit which can keep state information
Encode - Decode Paradigm Explained

Use two RNN networks: the encoder and the decoder

(1) Encoder processes one sequence at a time

(2) A representation of the sentence is generated

(3) This is used to initialize the decoder state

(4) Decoder generates one element at a time

(5) … continue till end of sequence tag is generated

Encoding

Decoding
This approach reduces the entire sentence representation to a single vector

Two problems with this design choice:

- A single vector is not sufficient to represent to capture all the syntactic and semantic complexities of a sentence
  - Solution: Use a richer representation for the sentences
- Problem of capturing long term dependencies: The decoder RNN will not be able to make use of source sentence representation after a few time steps
  - Solution: Make source sentence information when making the next prediction
  - Even better, make RELEVANT source sentence information available

These solutions motivate the next paradigm
**Encode - Attend - Decode Paradigm**

Represent the source sentence by the set of output vectors from the encoder.

Each output vector at time \( t \) is a contextual representation of the input at time \( t \).

*Note: in the encoder-decode paradigm, we ignore the encoder outputs*.

Let’s call these encoder output vectors *annotation vectors*.

---

**Diagram:**

- **s₀**: Input
- **s₁**: “read”
- **s₂**: “the”
- **s₃**: “book”
- **o₁**, **o₂**, **o₃**, **o₄**: Output vectors

---

**Annotations:**

- **Read**: “I read the book”
How should the decoder use the set of annotation vectors while predicting the next character?

Key Insight:
(1) Not all annotation vectors are equally important for prediction of the next element
(2) The annotation vector to use next depends on what has been generated so far by the decoder

eg. To generate the 3\textsuperscript{rd} target word, the 3\textsuperscript{rd} annotation vector (hence 3\textsuperscript{rd} source word) is most important

One way to achieve this:
Take a weighted average of the annotation vectors, with more weight to annotation vectors which need more focus or attention

This averaged context vector is an input to the decoder
Let’s see an example of how the attention mechanism works during decoding.

For generation of $i^{th}$ output character:

- $c_i$: context vector
- $a_{ij}$: annotation weight for the $j^{th}$ annotation vector
- $o_j$: $j^{th}$ annotation vector

\[
c_i = \sum_{j=1}^{n} a_{ij} o_j
\]
मैंने किताब पढ़ी <EOS>
But we do not know the attention weights? How do we find them?

*Let the training data help you decide!!*

**Idea:** Pick the attention weights that maximize the translation accuracy *more precisely, decrease training data loss*

- Note ⇒ *no separate language model*
- Neural MT generates fluent sentences
- Quality of word order is better
- No combinatorial search required for evaluating different word orders:
  - Decoding is very efficient compared to PBSMT
- *Exciting times ahead!*
We can look at translation as a sequence to sequence transformation problem

Read the entire sequence and predict the output sequence (using function $F$)

- Length of output sequence need not be the same as input sequence
- Prediction at any time step $t$ has access to the entire input
- A very general framework
Sequence to Sequence transformation is a very general framework

Many other problems can be expressed as sequence to sequence transformation

- **Summarization**: Article $\Rightarrow$ Summary
- **Question answering**: Question $\Rightarrow$ Answer
- **Image labelling**: Image $\Rightarrow$ Label
- **Transliteration**: character sequence $\Rightarrow$ character sequence
Summary

- Introduction
- Machine Translation Paradigms
- Phrase-based SMT
- Extensions to Phrase-based SMT
- Evaluation of Machine Translation
- Neural Machine Translation
Getting Started with Machine Translation

Software

- Statistical Machine Translation: Moses
- Neural Machine Translation: Nematus

Corpora

- Technology Development in Indian Language (TDIL) website
- Europarl Corpus
- IIT Bombay English-Hindi Parallel Corpus
Resources for Reading

Books & Articles
- Statistical MT Tutorial Workbook, Kevin Knight (online)
- Statistical Machine Translation, Phillip Koehn (book)
- Machine Translation. Pushpak Bhattacharyya (book)
- Neural Machine Translation. Kyunghyun Cho (online)

Presentations
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
<table>
<thead>
<tr>
<th>Hindi</th>
<th>Punjabi Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>यदि श्वास प्रणालिका में सूजन आ जाये तब भी रक्त मुँह के रास्ते बाहर आने लगता है ।</td>
<td>नेत्रत मात भूरुणी तित्तृ मैं आग जाने उठ ती बुझ मूंह दे तमारे घांट भरीठ छोड़ा ता । । जेकर साह प्रणाली विच सोज आ जावे तद वी खून मूँह दे रास्ते बाहर आउण लगदा है ।</td>
</tr>
<tr>
<td>Hindi-Punjabi Transliteration</td>
<td></td>
</tr>
<tr>
<td>आचि मात भूरुणी में मुँड़ठ आग जाने उठ ती बुझ मूँह दे तमारे घांट भरीठ छोड़ा ता । । आदि साह प्रणाली में सूजन आ जावे तद वी रक्त मूँह दे रास्ते बाहर आउण लगदा है ।</td>
<td></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
What is transliteration?

- 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
    - नरेंद्र → नरेंदर (नरेंदर)
  - 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
  - കോഴിക്കോട് (kozhikkoT)→ कोषिक्कोड (koShikkod)
- Transliterate or translate
  - കോഴിക്കോട് (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

Remember the **varnamala**? – organized according to scientific principles
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
● Hands on
  – Google Transliterate
    ● http://www.google.com/inputtools/
  – Brahmi-Net: IITB's transliteration system
    ● http://www.cfilt.iitb.ac.in/brahminet/
Thank You!

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