Machine Translation

Anoop Kunchukuttan
Microsoft, MT Group, Hyderabad
anoop.kunchukuttan@gmail.com

Summer School on Natural Language Processing Hosted by IIIT Hyderabad (online), 5 to 16 July 2021

Outline

- **Introduction**
- Statistical Machine Translation
- Neural Machine Translation
- Evaluation of Machine Translation
- Multilingual Neural Machine Translation

Deeper Outline of NMT Topics to cover

- *Why NMT?*
- *Encoder-Decoder Models*
- *Attention Mechanism*
- *Transformer Networks*
- *Backtranslation*
- *Subword-level Models*
- *Advanced Seq2Seq Modeling*

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

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

What is Machine Translation?

<u>Word order: SOV (Hindi), SVO (English)</u>

E: $\overbrace{\text{Germany}}^{S}\ \overbrace{\text{won}}^{V}\ \text{the last}\ \overbrace{\text{World Cup}}^{O}$

H: $\underbrace{\text{जर्मनी}}_{S}\ \text{ने}\ \text{पिछला}\ \underbrace{\text{विश्व कप}}_{O}\ \underbrace{\text{जीता था}}_{V}$

<u>Free (Hindi) vs rigid (English) word order</u>

पिछला विश्व कप जर्मनी ने जीता था *(correct)*

The last World Cup Germany won *(grammatically incorrect)*
The last World Cup won Germany *(meaning changes)*

Language Divergence ➔ the great diversity among languages of the world

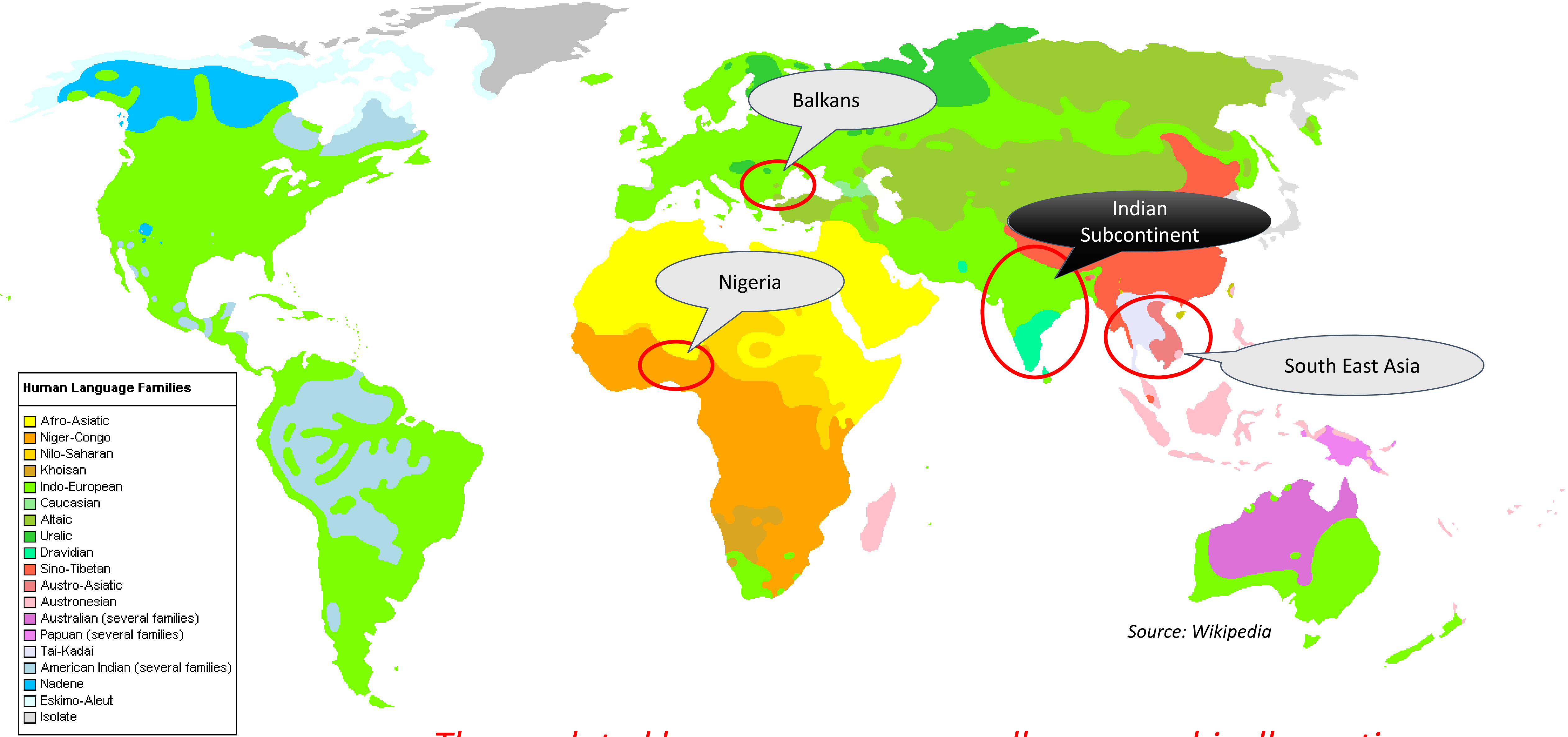
The central problem of MT is to bridge this language divergence

Shachi Dave, Jignashu Parikh and Pushpak Bhattacharyya, Interlingua Based English Hindi Machine Translation and Language Divergence, Journal of Machine Translation (JMT), Volume 17, September, 2002.

Why is Machine Translation interesting?

Language Divergence → *the great diversity among languages of the world*

The central problem of MT is to bridge this language divergence



These related languages are generally geographically contiguous

Related Languages

Related by Genealogy

<u>Language Families</u>

Dravidian, Indo-European, Turkic

(Jones, Rasmus, Verner, 18th & 19th centuries, Raymond ed. (2005))

Related by Contact

<u>Linguistic Areas</u>

Indian Subcontinent, Standard Average European

(*Trubetzkoy, 1923*)

Related languages may not belong to the same language family!

Language Divergence

<u>Word order: SOV (Hindi), SVO (English), VSO, OSV</u>

S V O

E: Germany won the last World Cup

H: जर्मनी ने पिछला विश्व कप जीता था

S O V

<u>Free (Hindi) vs rigid (English) word order</u>

पिछला विश्व कप जर्मनी ने जीता था *(correct)*

The last World Cup Germany won *(grammatically incorrect)*

The last World Cup won Germany *(meaning changes)*

Language Divergence

Analytic vs Polysynthetic languages

Analytic (Chinese) → very few morphemes per word, no inflections

Polysynthetic (Finnish)→ many morphemes per word, no inflections

English: *Even if it does not rain*

Malayalam: മഴ പെയ്യുതിലെങ്ങിലും

(rain_noun shower_verb+not+even_if+then_also)

Inflectional systems [infixing (Arabic), fusional (Hindi), agglutinative (Marathi)]

Arabic

k-t-b: root word
katabtu: I wrote
kattabtu: I had (something) written
kitaab: book
kotub: books

Hindi

Jaaunga (1st per, singular, masculine)
Jaaoge (2nd per)
Jaayega (3rd per, singular, masculine)
Jaayenge (3rd per, plural)

Marathi

कपाटावरील: कपाट + वर + ईल
(the one over the cupboard)
दारावरील: दार + वर + ईल
(the one over the door)
दारामागील: दार + मागे + ईल
(the one behind the door)

Language Divergence

<u>Different ways of expressing same concept</u>

water → पानी, जल, नीर

<u>Language registers</u>

Formal: आप बैठिये

Informal: तू बैठ

Standard : मुझे डोसा चाहिए

Dakhini: मेरे को डोसा होना

Language Divergence

- Case marking systems
- Categorical divergence
- Null Subject Divergence
- Pleonastic Divergence

… and much more

Shachi Dave, Jignashu Parikh and Pushpak Bhattacharyya, Interlingua Based English Hindi Machine Translation and Language Divergence, Journal of Machine Translation (JMT), Volume 17, September, 2002.

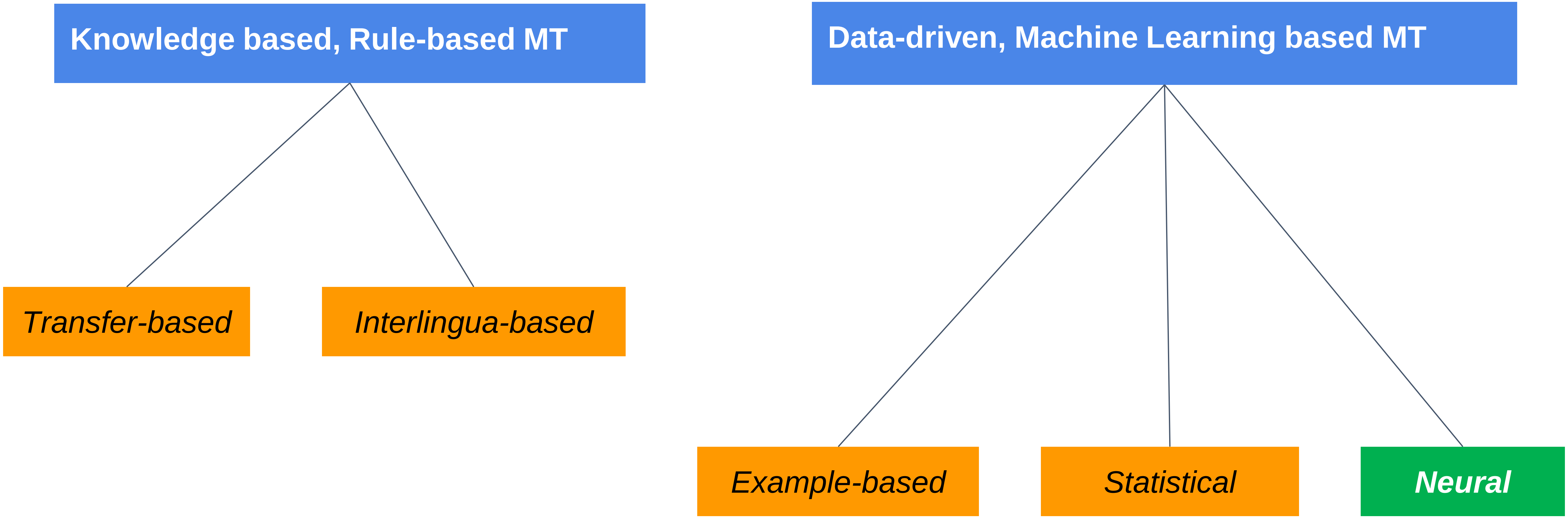
Why is Machine Translation difficult?

- **Ambiguity**
  - Same word, multiple meanings: *मंत्री (minister or chess piece)*
  - 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
  - *घर ा समोर चा*
  - *That which is in front of the house*

Why should you study Machine Translation?

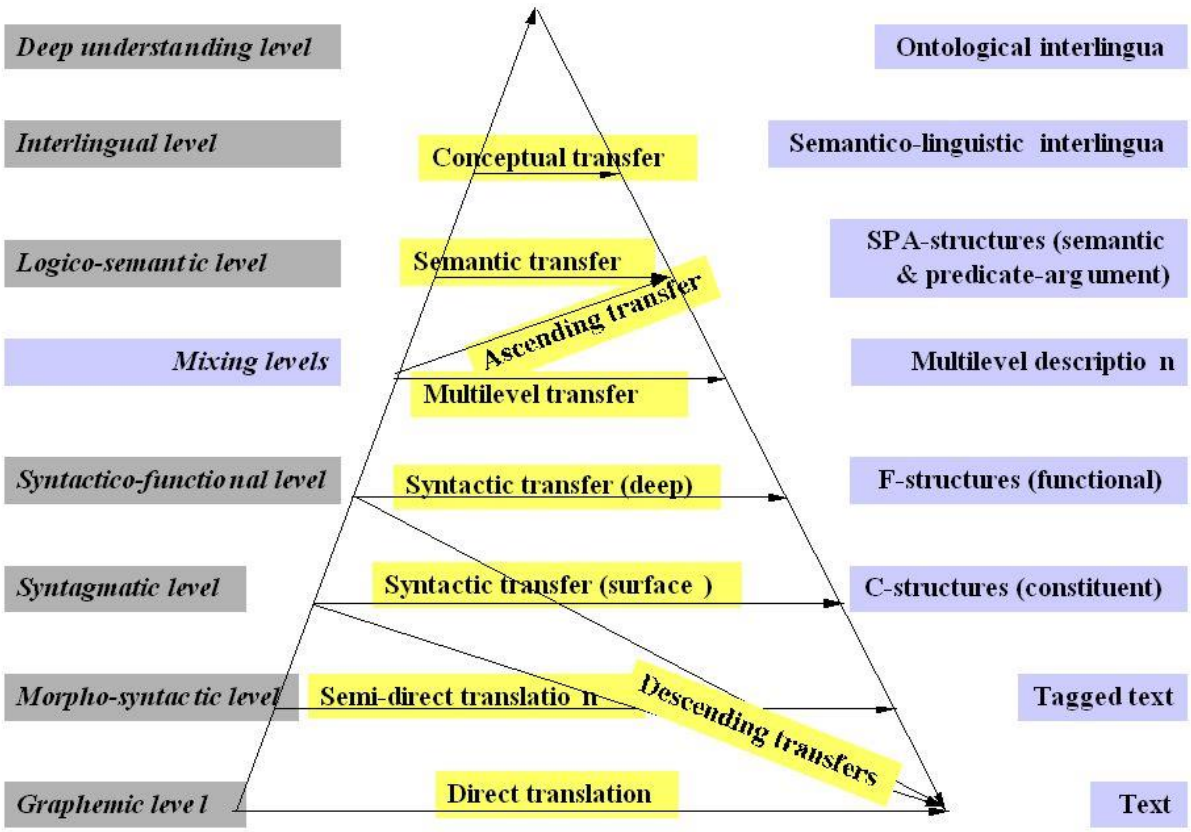
- One of the most challenging problems in Natural Language Processing
- Pushes the boundaries of NLP
- Involves analysis as well as synthesis
- Involves all layers of NLP: morphology, syntax, semantics, pragmatics, discourse
- *Theory and techniques in MT are applicable to a wide range of other problems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

Approaches to build MT systems



Vauquois Triangle

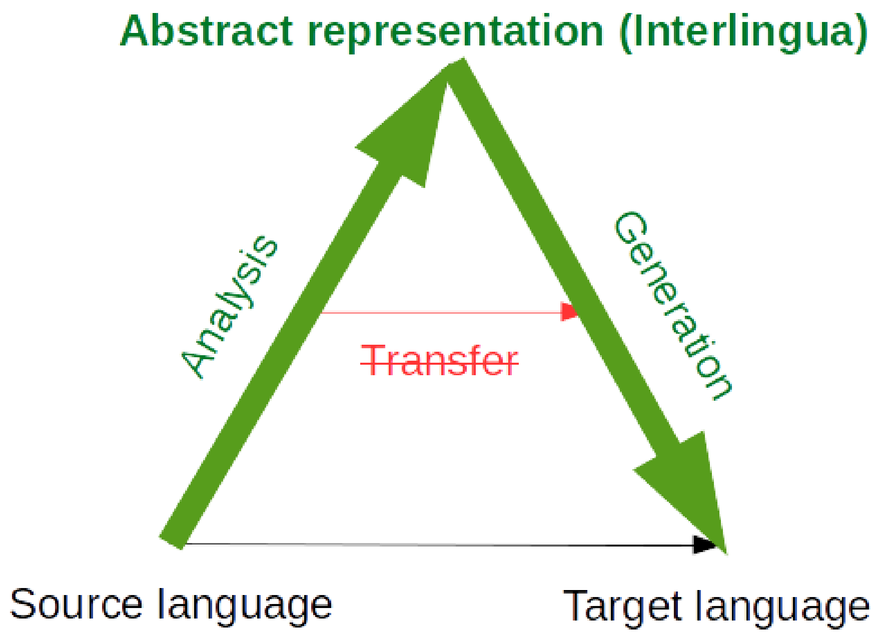
Translation approaches can be classified by the depth of linguistic analysis they perform



Rule-based MT

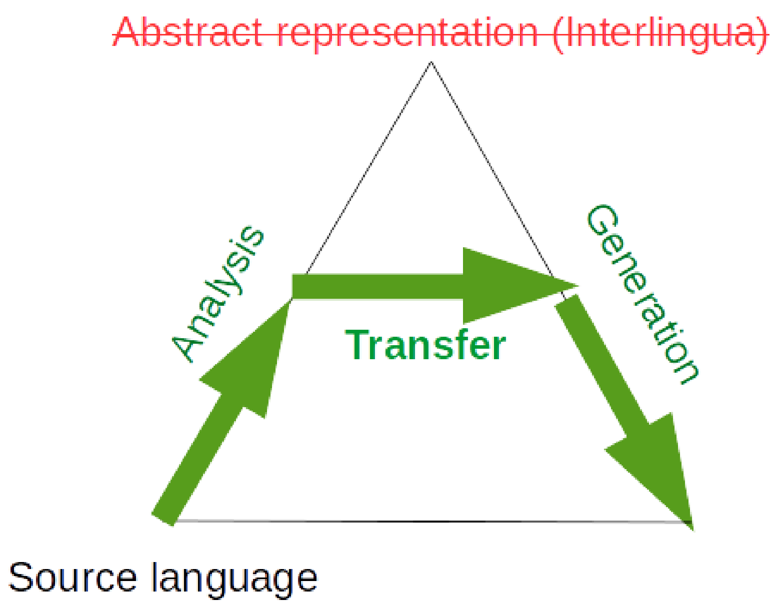
- Rules are written by ***linguistic experts*** to analyze the source, generate an intermediate representation, and generate the target sentence
- Depending on the depth of analysis: interlingua or transfer-based MT

Interlingua based MT



Deep analysis, complete disambiguation and language independent representation

Transfer based MT



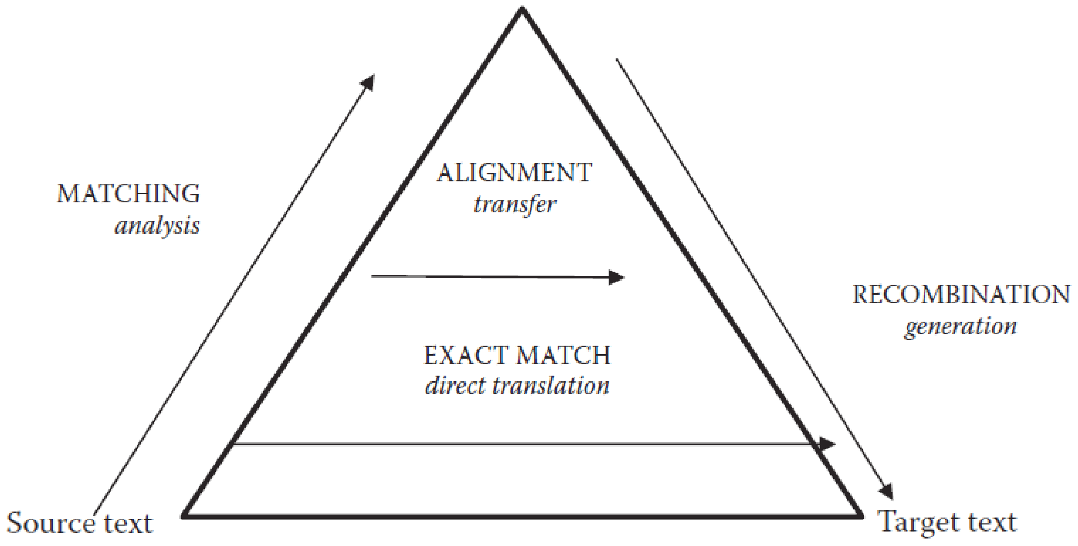
Partial analysis, partial disambiguation and a bridge intermediate representation

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)***

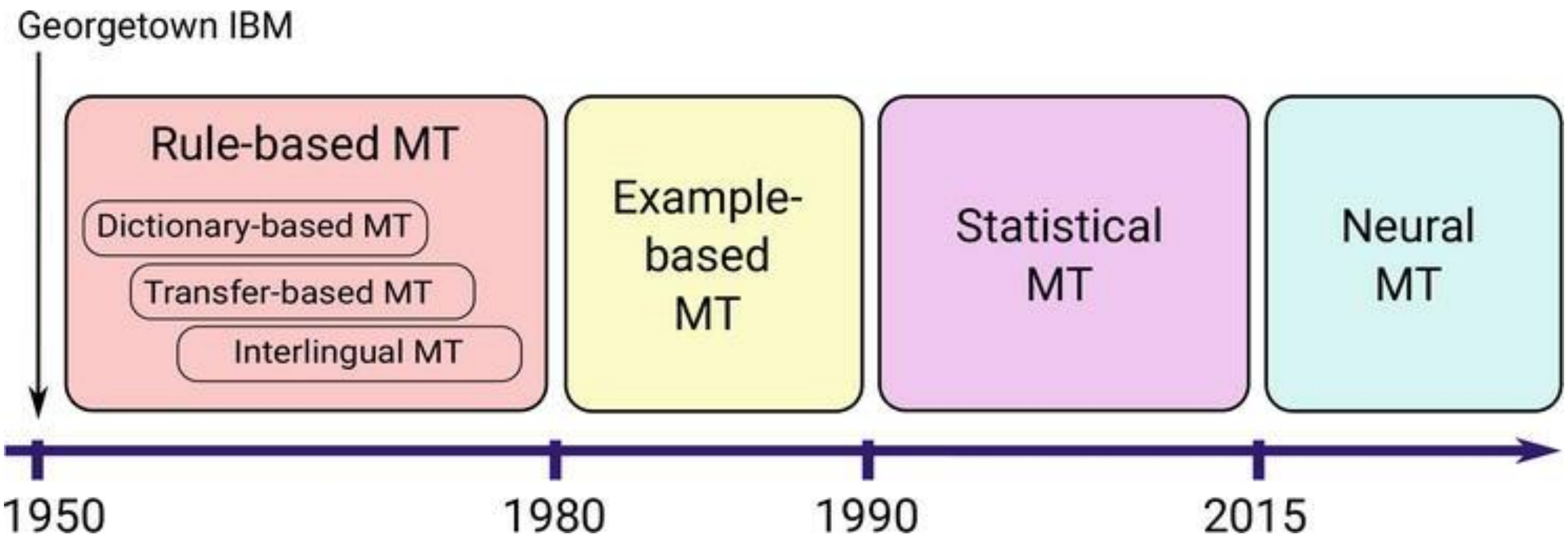
   वह खरीदता है
   एक किताब
   अंतर राष्ट्रीय राजनीति

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*

The Evolution of MT systems



Source: https://www.intechopen.com/books/recent-trends-in-computational-intelligence/machine-translation-and-the-evaluation-of-its-quality

Outline

- Introduction
- **Statistical Machine Translation**
- Neural Machine Translation
- Evaluation of Machine Translation
- Multilingual Neural Machine Translation

Statistical Machine Translation

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*

E: target language
F: source language

e: target language sentence
f : source language sentence

Best translation

$$\bar{e} = \arg\max_{e} P(e|f)$$

How do we **model** this quantity?

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

We must first explain the process of translation

A very general framework for many NLP problems

We explain translation using the Noisy Channel Model

Generate target sentence

Channel corrupts the target

Source sentence is a corruption of the target sentence

Language Model (LM)
P(e)

Captures fluency

Translation Model (TM)
P(f|e)

Captures fidelity

$$P(e|f) = P(e) \times P(f|e)$$

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

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

Brown, Peter F., et al. "The mathematics of statistical machine translation: Parameter estimation." *Computational linguistics* 19.2 (1993): 263-311.

We know how to learn n-gram language models

Let's see how to learn the translation model → $P(\boldsymbol{f}|\boldsymbol{e})$

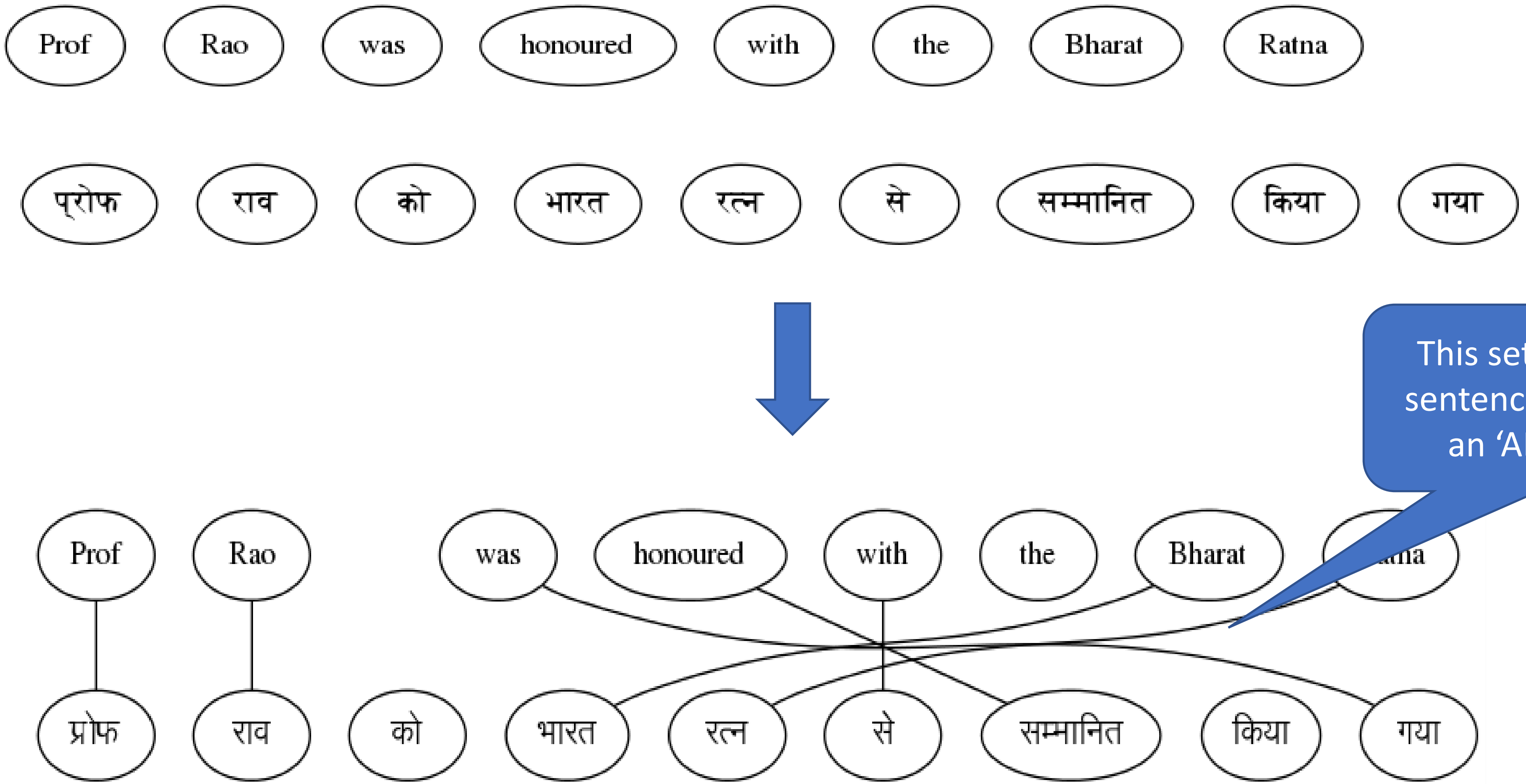
To learn sentence translation probabilities,

→ we first need to learn word-level translation probabilities

That is the task of word alignment

Parallel Corpus	
A boy is sitting in the kitchen	एक लडका रसोई मे बैठा है
A boy is playing tennis	एक लडका टेनिस खेल रहा है
A boy is sitting on a round table	एक लडका एक गोल मेज पर बैठा है
Some men are watching tennis	कुछ आदमी टेनिस देख रहे है
A girl is holding a black book	एक लडकी ने एक काली किताब पकडी है
Two men are watching a movie	दो आदमी चलचित्र देख रहे है
A woman is reading a book	एक औरत एक किताब पढ रही है
A woman is sitting in a red car	एक औरत एक काले कार मे बैठी है

Given a parallel sentence pair, find word level correspondences

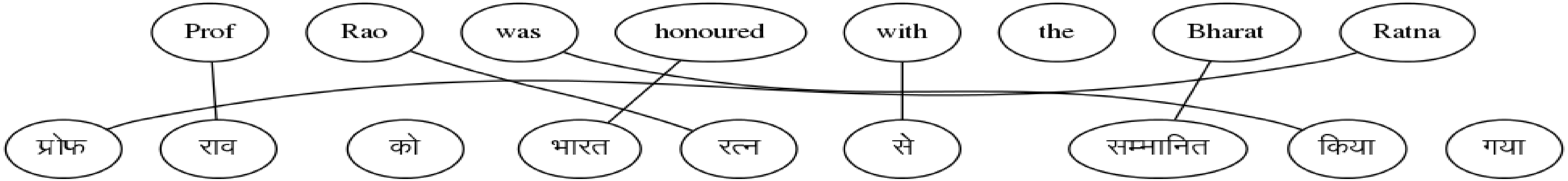


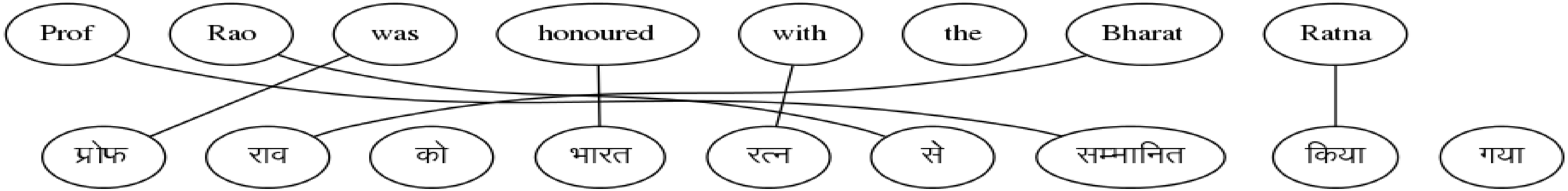
This set of links for a sentence pair is called an 'ALIGNMENT'

Brown, Peter F., et al. "The mathematics of statistical machine translation: Parameter estimation." *Computational linguistics* 19.2 (1993): 263-311.

But there are multiple possible alignments

Sentence 1

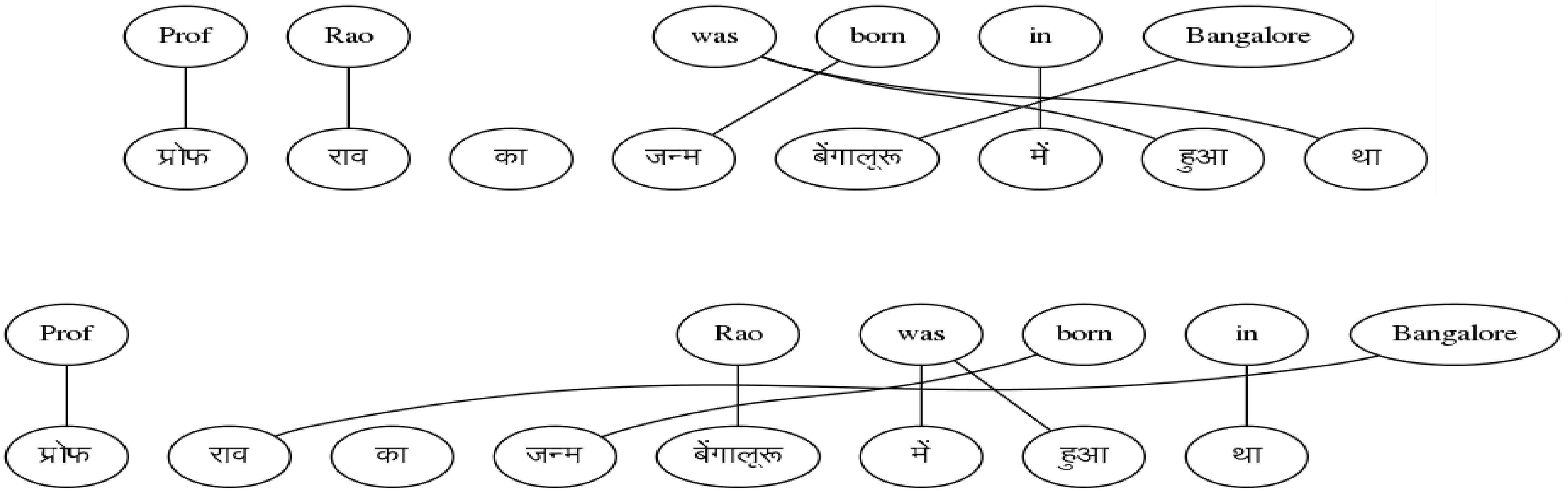




With one sentence pair, we cannot find the correct alignment

Can we find alignments if we have multiple sentence pairs?

Sentence 2



Yes, let’s see how to do that …

Parallel Corpus	
A boy is **sitting** in the kitchen	एक लडका रसोई मे **बैठा** है
A boy is playing **tennis**	एक लडका **टेनिस** खेल रहा है
A boy is **sitting** on a round table	एक लडका एक गोल मेज पर **बैठा** है
Some men **are watching** **tennis**	कुछ आदमी **टेनिस** **देख रहे है**
A girl is holding a black book	एक लडकी ने एक काली किताब पकडी है
Two men **are watching** a movie	दो आदमी चलचित्र **देख रहे है**
A woman is reading a book	एक औरत एक किताब पढ रही है
A woman is **sitting** in a red car	एक औरत एक काले कार मे **बैठा** है

https://www.isi.edu/natural-language/mt/wkbk.rtf

Key Idea

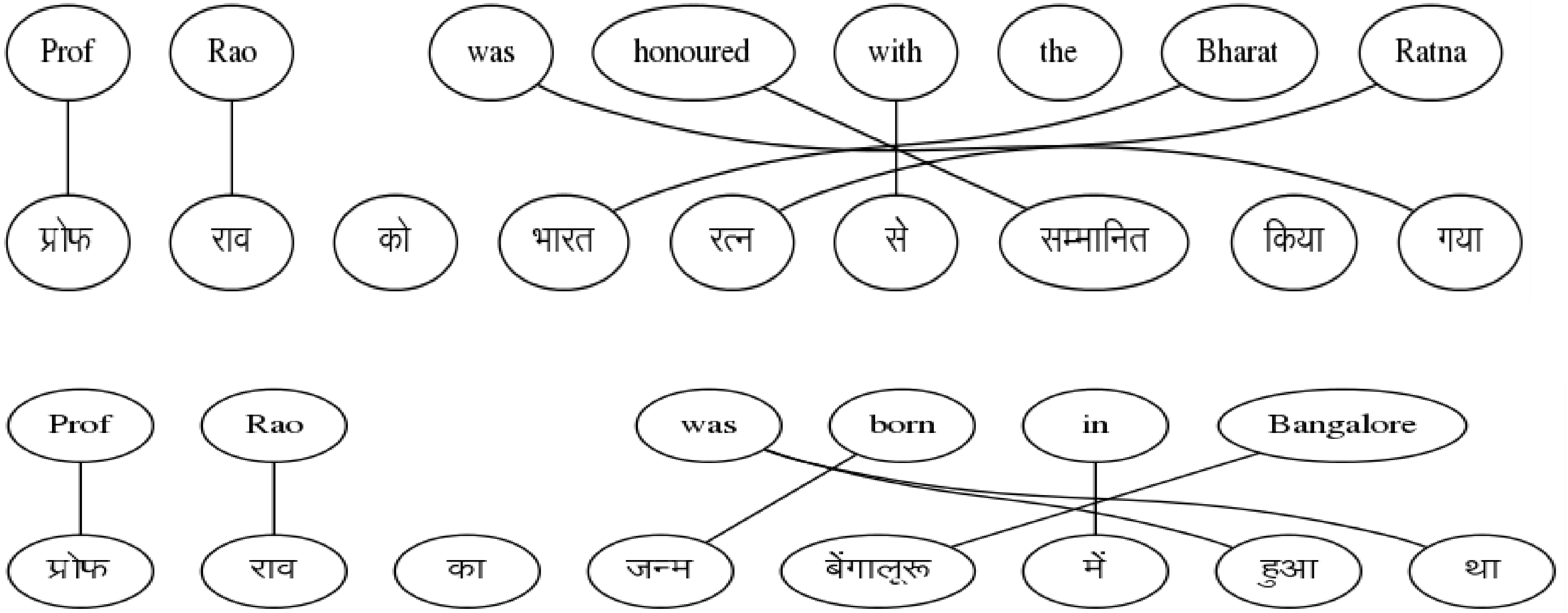
Co-occurrence of translated words

Words which occur together in the parallel sentence are likely to be translations (higher P(f|e))

If we knew the alignments, we could compute P(f|e)

Sentence 1

Sentence 2



$$P(f|e) = \frac{\#(f,e)}{\#(*,e)}$$

$$P(Prof|\text{प्रोफ}) = \frac{2}{2}$$

$\#(a,b)$: *number of times word a is aligned to word b*

But, we can find the best alignment only if we know the word translation probabilities

The best alignment is the one that maximizes the sentence translation probability

$$P(\boldsymbol{f},\boldsymbol{a}|\boldsymbol{e}) = P(a)\prod_{i=1}^{i=m} P\left(f_i|e_{a_i}\right) \quad \Longrightarrow \quad \boldsymbol{a}^* = \underset{\boldsymbol{a}}{\operatorname{argmax}} \prod_{i=1}^{i=m} P(f_i|e_{a_i})$$

This is a chicken and egg problem! How do we solve this?

We can solve this problem using a two-step, iterative process

Start with random values for word translation probabilities

Step 1: Estimate alignment probabilities using word translation probabilities

Step 2: Re-estimate word translation probabilities

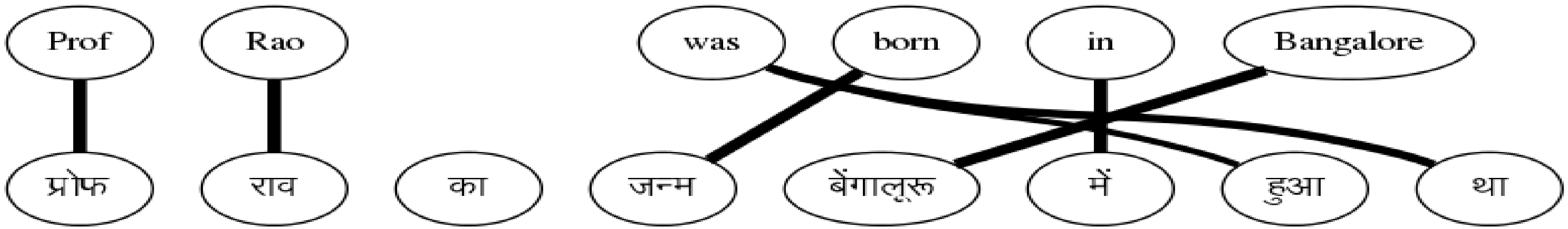
- *We don't know the best alignment*
- *So, we consider all alignments while estimating word translation probabilities*
- *Instead of taking only the best alignment, we consider all alignments and weigh the word alignments with the alignment probabilities*

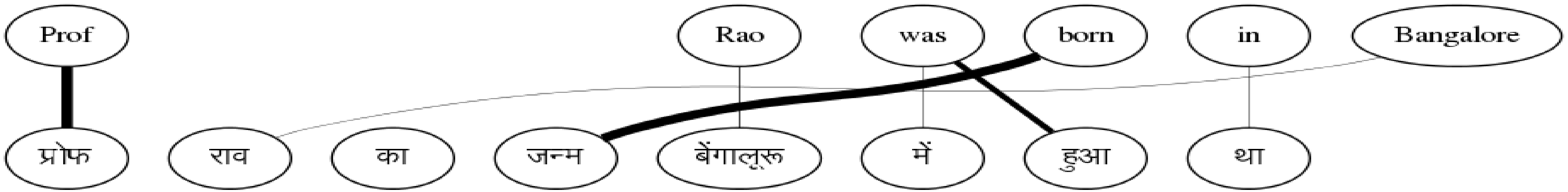
$$P(f|e) = \frac{expected\ \#(f,e)}{expected\ \#(*,e)}$$

Repeat Steps (1) and (2) till the parameters converge

At the end of the process …

Sentence 2





Expectation-Maximization Algorithm: *guaranteed to converge, maybe to local minima*
Hence we need to good initialization and training regimens.

IBM Models

- IBM came up with a series of increasingly complex models
- Called Models 1 to 5
- Differed in assumptions about alignment probability distributions
- Simper models are used to initialize the more complex models
- This pipelined training helped ensure better solutions

Phrase Based SMT

Why stop at learning word correspondences?

KEY IDEA ➔ Use "Phrase" (Sequence of Words) as the basic translation unit

Note: the term 'phrase' is not used in a linguistic sense

The Prime Minister of India	भारत के प्रधान मंत्री bhArata ke pradhAna maMtrI India of Prime Minister
is running fast	तेज भाग रहा है teja bhAg rahA hai fast run -continuous is
honoured with	से सम्मानित किया se sammanita kiyA with honoured did
Rahul lost the match	राहुल मुकाबला हार गया rAhula mukAbalA hAra gayA Rahul match lost

Koehn, Philipp, Franz J. Och, and Daniel Marcu. Statistical phrase-based translation. UNIVERSITY OF SOUTHERN CALIFORNIA MARINA DEL REY INFORMATION SCIENCES INST, 2003. https://apps.dtic.mil/sti/pdfs/ADA461156.pdf

Benefits of PB-SMT

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

The Prime Minister of India	भारत के प्रधान मंत्री bhaarat ke pradhaan maMtrI India of Prime Minister

Sense disambiguation based on local context → Neighbouring words help make the choice

heads towards Pune	पुणे की ओर जा रहे है pune ki or jaa rahe hai Pune towards go –continuous is
heads the committee	समिति की अध्यक्षता करते है Samiti kii adhyakshata karte hai committee of leading -verbalizer is

Benefits of PB-SMT (2)

Handling institutionalized expressions

- Institutionalized expressions, idioms can be learnt as a single unit

hung assembly	त्रिशंकु विधानसभा trishanku vidhaansabha
Home Minister	गृह मंत्री gruh mantrii
Exit poll	चुनाव बाद सर्वेक्षण chunav baad sarvekshana

- Improved Fluency
  - The phrases can be arbitrarily long (even entire sentences)

Parallel Corpus	
A boy is **sitting** in the kitchen	एक लडका रसोई मे **बैठा** है
A boy is playing **tennis**	एक लडका **टेनिस** खेल रहा है
A boy is **sitting** on a round table	एक लडका एक गोल मेज पर **बैठा** है
Some men **are watching tennis**	कुछ आदमी **टेनिस देख रहे है**
A girl is holding a black book	एक लडकी ने एक काली किताब पकडी है
Two men **are watching** a movie	दो आदमी चलचित्र **देख रहे है**
A woman is reading a book	एक औरत एक किताब पढ रही है
A woman is **sitting** in a red car	एक औरत एक काले कार मे **बैठा** है

Mathematical Model

$$\mathbf{e}_{\text{best}} = \text{argmax}_{\mathbf{e}}\, p(\mathbf{e}|\mathbf{f}) = \text{argmax}_{\mathbf{e}}\, p(\mathbf{f}|\mathbf{e})\, p_{\text{LM}}(\mathbf{e})$$

<u>Let's revisit the decision rule for SMT model</u>

<u>Let's revisit the translation model *p*(**f|e**)</u>

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

$$p(\bar{f}_1^I|\bar{e}_1^I) = \prod_{i=1}^{I} \phi(\bar{f}_i|\bar{e}_i)\, d(\text{start}_i - \text{end}_{i-1} - 1)$$

Distortion probability

Phrase Translation Probability

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

Learning The Phrase Translation Model

Involves Structure + Parameter Learning:

- Learn the **Phrase Table**: the central data structure in PB-SMT

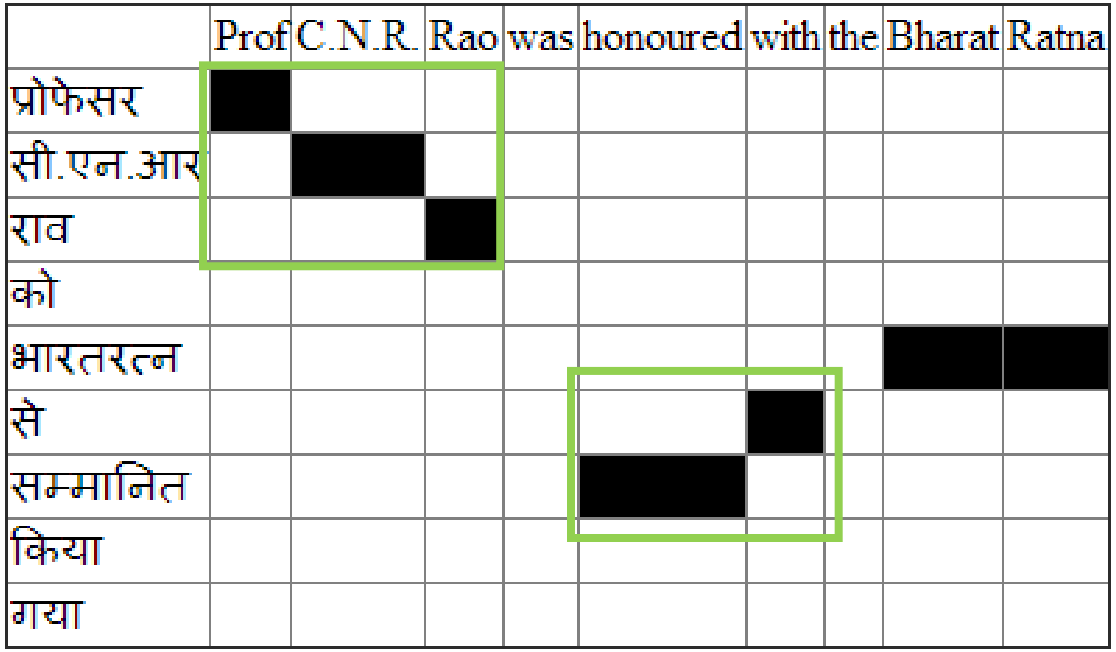
The Prime Minister of India	भारत के प्रधान मंत्री
is running fast	तेज भाग रहा है
the boy with the telescope	दूरबीन से लड़के को
Rahul lost the match	राहुल मुकाबला हार गया

- Learn the **Phrase Translation Probabilities**

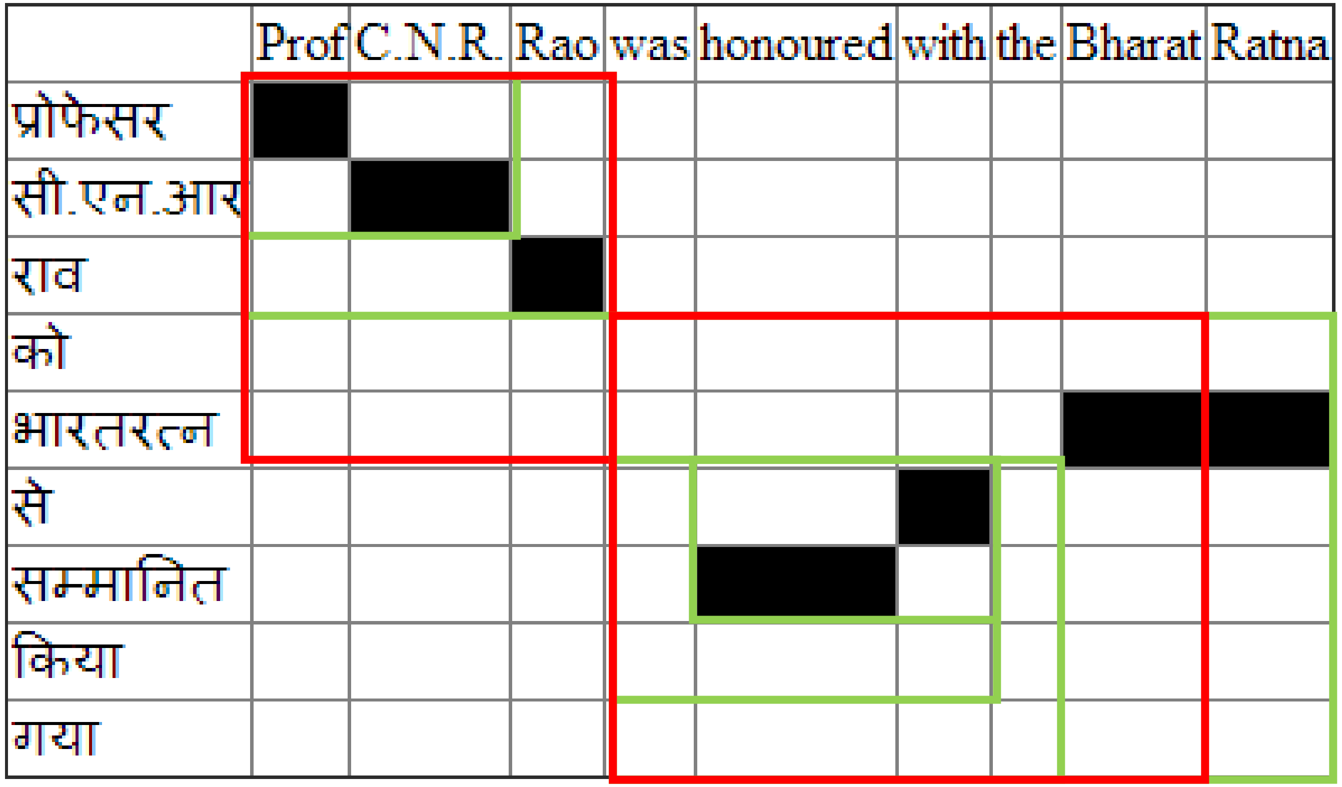
Prime Minister of India	भारत के प्रधान मंत्री India of Prime Minister	0.75
Prime Minister of India	भारत के भूतपूर्व प्रधान मंत्री India of former Prime Minister	0.02
Prime Minister of India	प्रधान मंत्री Prime Minister	0.23

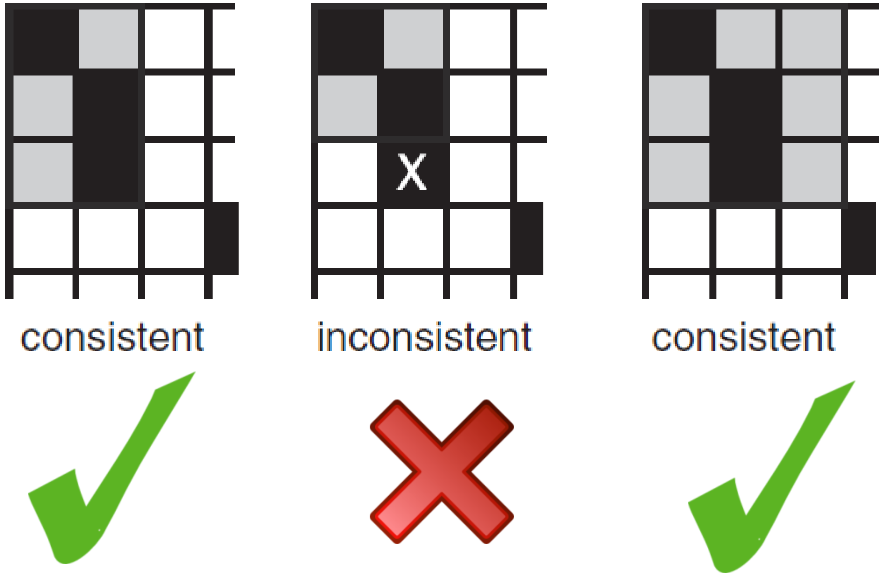
Learning Phrase Tables from Word Alignments

- Start with word alignments
- 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"



Which phrase pairs to include in the phrase table?





Source: SMT, Phillip Koehn

Professor CNR	प्रोफेसर सी.एन.आर
Professor CNR Rao	प्रोफेसर सी.एन.आर राव
Professor CNR Rao was	प्रोफेसर सी.एन.आर राव
Professor CNR Rao was	प्रोफेसर सी.एन.आर राव को
honoured with the Bharat Ratna	भारतरत्न से सम्मानित
honoured with the Bharat Ratna	भारतरत्न से सम्मानित किया
honoured with the Bharat Ratna	भारतरत्न से सम्मानित किया गया
honoured with the Bharat Ratna	को भारतरत्न से सम्मानित किया गया

Discriminative Training of PB-SMT

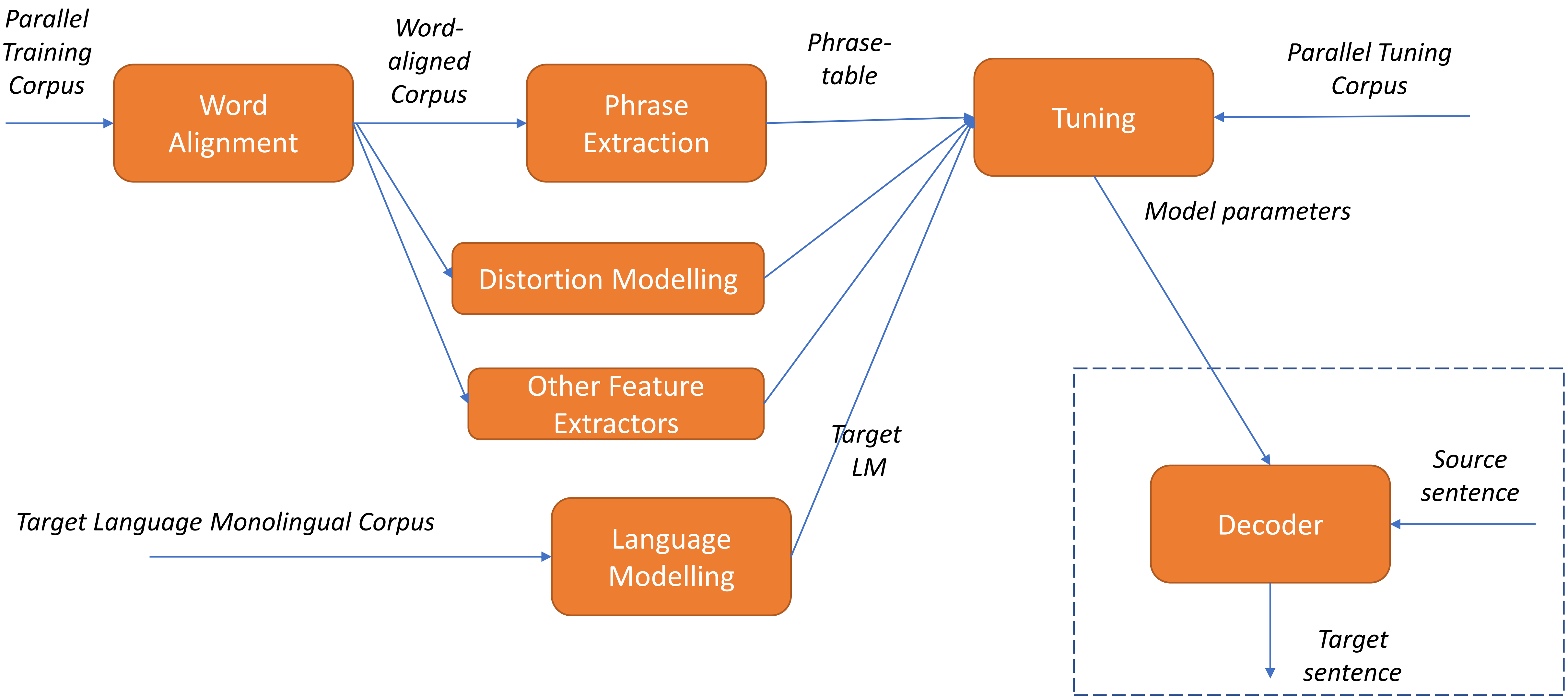
- Directly model the posterior probability p**(e|f)**
- Use the Maximum Entropy framework

$$P(\mathbf{e}|\mathbf{f}) = \exp\left(\sum_i \lambda_i h_i(f_1^I, e_1^J)\right)$$

$$e^* = \arg\max_{e_i} \sum_i \lambda_i h_i(f_1^I, e_1^J)$$

  - h_i**(f,e)** are feature functions , λ_i's are feature weights
- Benefits:
  - *Can add arbitrary features to score the translations*
  - Can assign different weight for each features
  - Assumptions of generative model may be incorrect
  - Feature weights λ_i are learnt during tuning

Typical SMT Pipeline



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

- Divergent Word Order
- Rich morphology
- Named Entities and Out-of-Vocabulary words

Limitations of SMT

- *No end-to-end optimization*
  - *Separately developed complex components strung together*
- *Divergent word-order is a big challenge*
- *n-gram LM not the best way to score translation fluency*
- *Model size is a function of the data size*

Outline

- Introduction
- Statistical Machine Translation
- **Neural Machine Translation**
- Evaluation of Machine Translation
- Multilingual Neural Machine Translation
- Summary

Neural Machine Translation

Topics

- *Why NMT?*
- *Encoder-Decoder Models*
- *Attention Mechanism*
- *Transformer Networks*
- *Backtranslation*
- *Subword-level Models*
- *Advanced Seq2Seq Modeling*

SMT, Rule-based MT and Example based MT **manipulate symbolic representations of knowledge**

<u>*Every word has an atomic representation, which can't be further analyzed*</u>

home	0
water	1
house	2
tap	3

1	0	0	0

0	1	0	0

0	0	1	0

0	0	0	1

No notion of similarity or relationship between words

- *Even if we know the translation of* `home`*, we can't translate* `house` *if it an OOV*

Difficult to represent new concepts

- *We cannot say anything about 'mansion' if it comes up at test time*
- *Creates problems for language model as well ⇒ whole are 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

<u>Every word is represented by a vector of numbers</u>

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

home	0.5	0.6	0.7
Water	0.2	0.9	0.3
house	0.55	0.58	0.77
tap	0.24	0.6	0.4

Word vectors or embeddings

Can define similarity between words

- *Vector similarity measures like cosine similarity*
- *Since* representations of `home` *and* `house`, *we may be able to translate* `house`

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*

Topics

- *Why NMT?*
- *Encoder-Decoder Models*
- *Attention Mechanism*
- *Transformer Networks*
- *Backtranslation*
- *Subword-level Models*
- *Advanced Seq2Seq Modeling*

Sequence Labelling Task

Input Sequence: $(x_1\ \ x_2\ \ x_3\ \ x_4 \ldots\ldots x_i \ldots\ldots x_N)$

Output Sequence: $(y_1\ \ y_2\ \ y_3\ \ y_4 \ldots\ldots y_i \ldots\ldots y_N)$

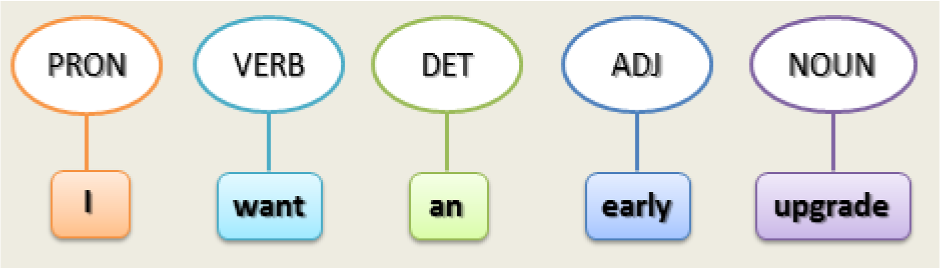
Input and output sequences have the same length

Variable length input

Output contains categorical labels

Output at any time-step typically depends on neighbouring output labels and input elements

Part-of-speech tagging



Recurrent Neural Network is a powerful model to learn sequence labelling tasks

Sequence to Sequence Task

Input Sequence: $(x_1 \;\; x_2 \;\; x_3 \;\; x_4 \ldots\ldots x_i \ldots\ldots x_N)$

Output Sequence: $(y_1 \;\; y_2 \;\; y_3 \;\; y_4 \ldots y_k \ldots y_M)$

Input and output sequences have different lengths

Variable length input

Output contains categorical labels

Output at any time-step typically depends on neighbouring output labels and input elements

Machine Translation

Encoder-decoder model is a general framework for sequence to sequence tasks

Many tasks as Sequence to Sequence transformations

- *Summarization: Article ⇒ Summary*
- *Question answering: Question ⇒ Answer*
- *Dialogue: Previous utterance ⇒ next utterance*
- *Transliteration: character sequence ⇒ character sequence*
- *Grammar Correction: Incorrect sentence ⇒ Correct Sentence*
- *Translation Postediting: Incorrect translation ⇒ Correct translation*
- *Image labelling: Image ⇒ Label*

We have seen what a language model is:

It models $P(\boldsymbol{Y})$

$$P(\boldsymbol{Y}) = \prod_{j=1}^{M} P(y_j | y_1, \underbrace{y_2, \ldots y_{j-1})}_{\textit{RNN}}$$

We are interested in modeling $P(\boldsymbol{Y}|\boldsymbol{X})$

Conditional Language Modelling task ➔

Learning Target LM conditioned on the source sentence

$$P(\boldsymbol{Y}|\boldsymbol{X}) = \prod_{j=1}^{M} P(y_j | \underbrace{y_1, y_2, \ldots y_{j-1}}_{\textit{Target language RNN}}, \boldsymbol{X})$$

Additional conditioning on the source sentence

How do we model this?

LM for generating the target sequence

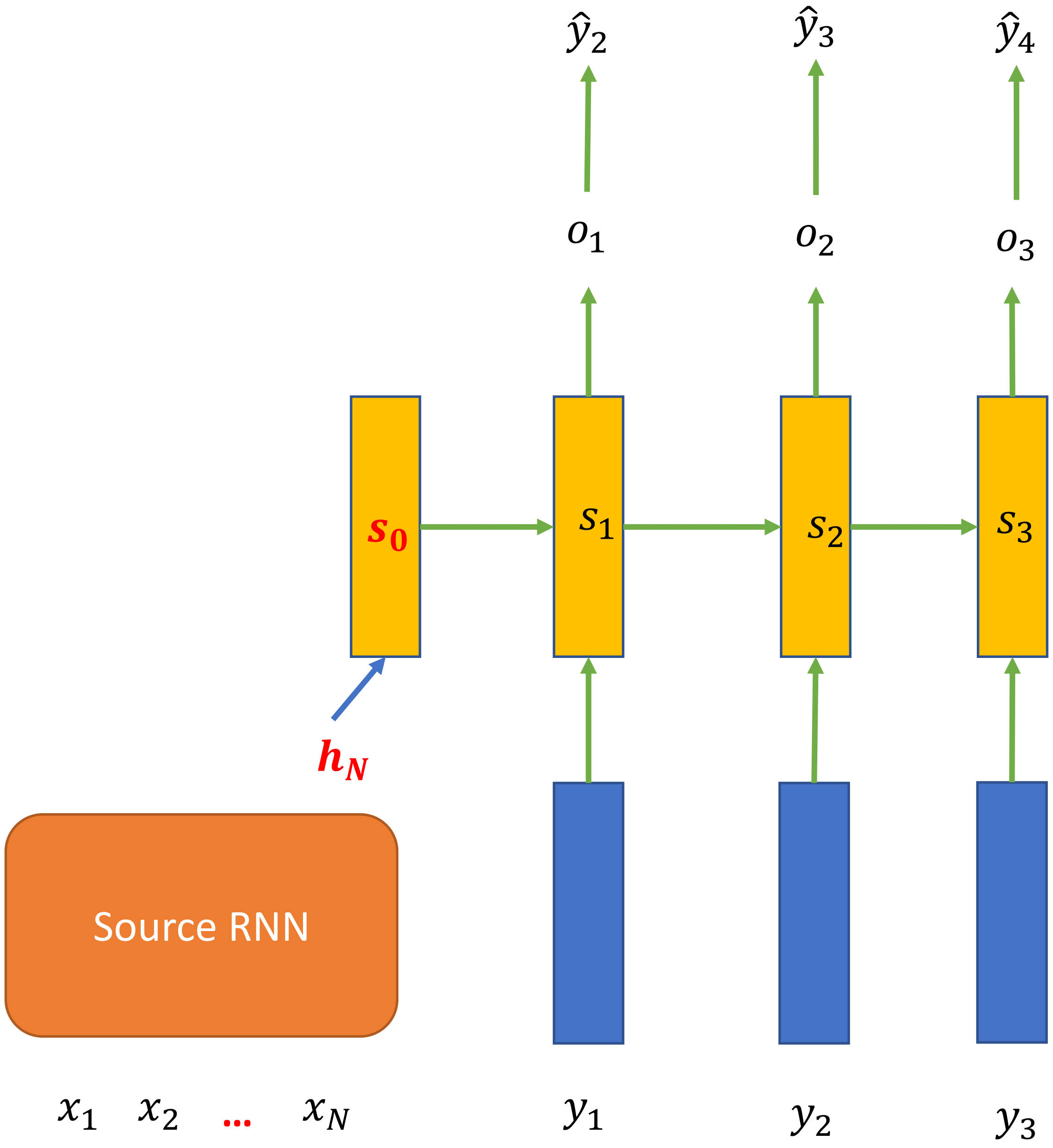
s_0 ➔ *Initial state of target language RNN*

Set s_0 = a vector representation of source

We have our ***conditional*** *LM*

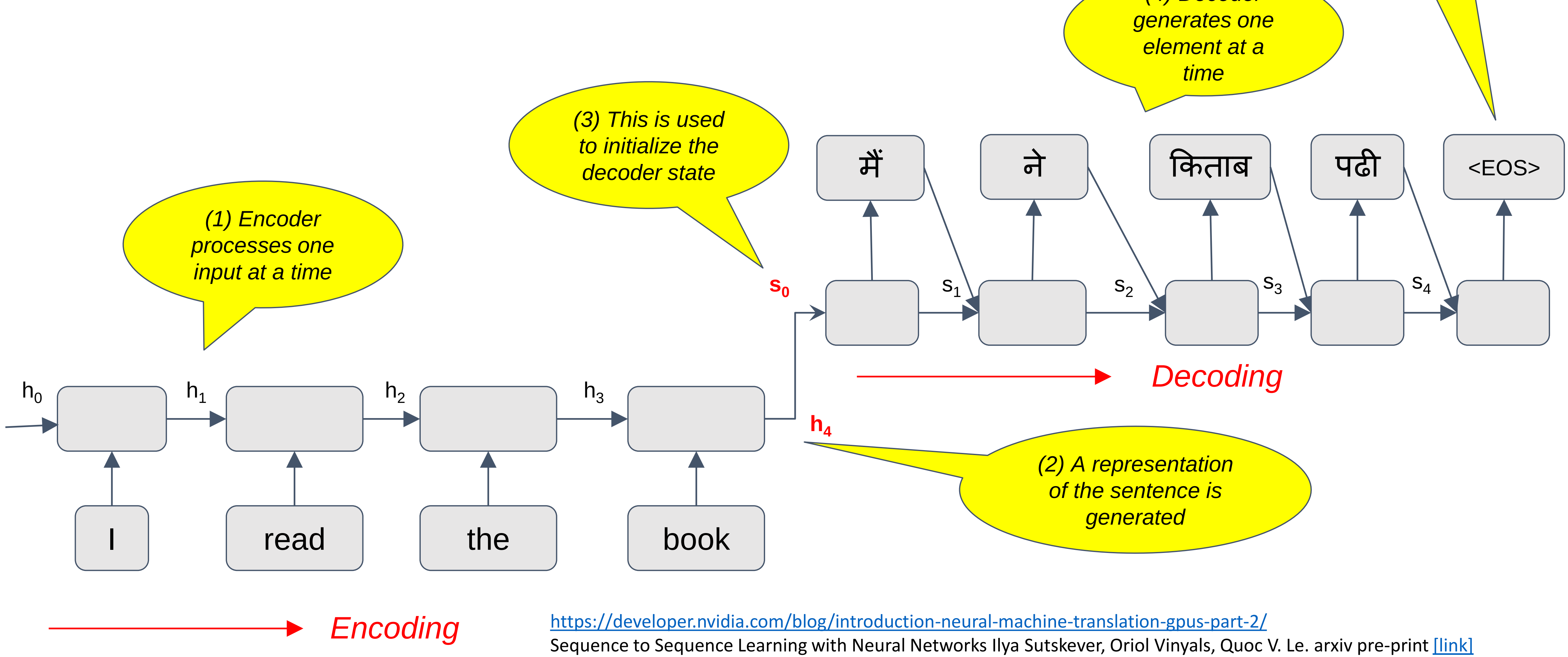
Source Vector Representation ➔
last state of source sentence RNN

$$s_0 = h_N$$



Encode - Decode Paradigm Explained

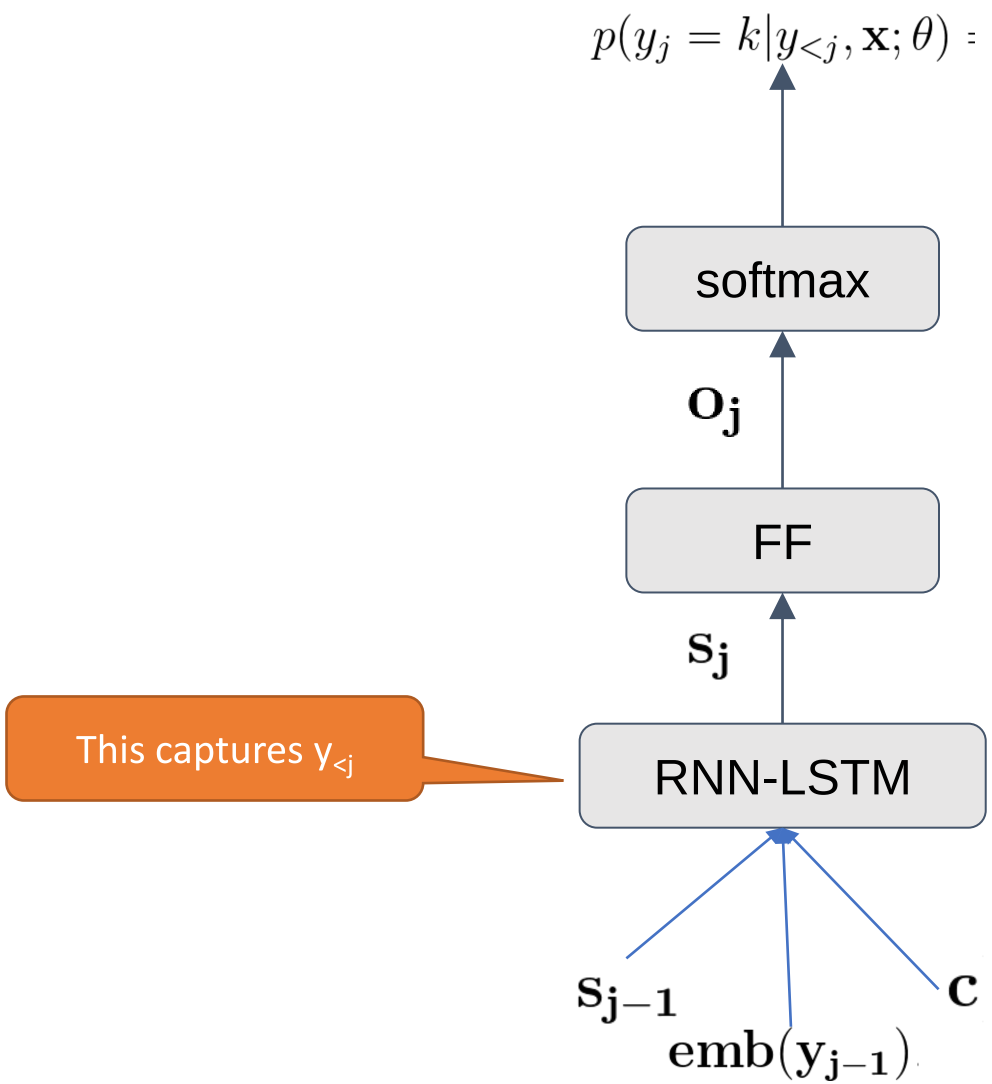
Use two RNN networks: the encoder and the decoder



https://developer.nvidia.com/blog/introduction-neural-machine-translation-gpus-part-2/

Sequence to Sequence Learning with Neural Networks Ilya Sutskever, Oriol Vinyals, Quoc V. Le. arxiv pre-print [link]

What is the decoder doing at each time-step?



$$softmax(o_{jk}) = \frac{\exp(o_{jk})}{\sum_{m=0}^{m=T} \exp(o_{jm})}$$

$$\mathbf{o_j} = FF(s_j)$$

$$\mathbf{s_j} = g(\mathbf{s_{j-1}}, \mathbf{emb(y_{j-1})}, \mathbf{c})$$

This captures x, $c=h_4$

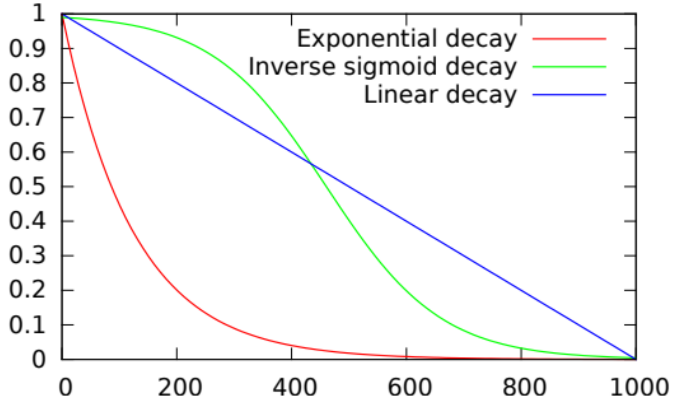
Training an NMT Model

$$p(\mathbf{y}|\mathbf{x};\theta) = \prod_{j=1}^{m} p(y_j|y_{<j},\mathbf{x};\theta) \qquad p(y_j = k|y_{<j},\mathbf{x};\theta) = softmax(o_{jk})$$

$$\mathcal{L}_\theta = \sum_{(\mathbf{x},\mathbf{y})\in\mathbf{C}} \log p(\mathbf{y}|\mathbf{x};\theta)$$

Maximum Likelihood Estimation

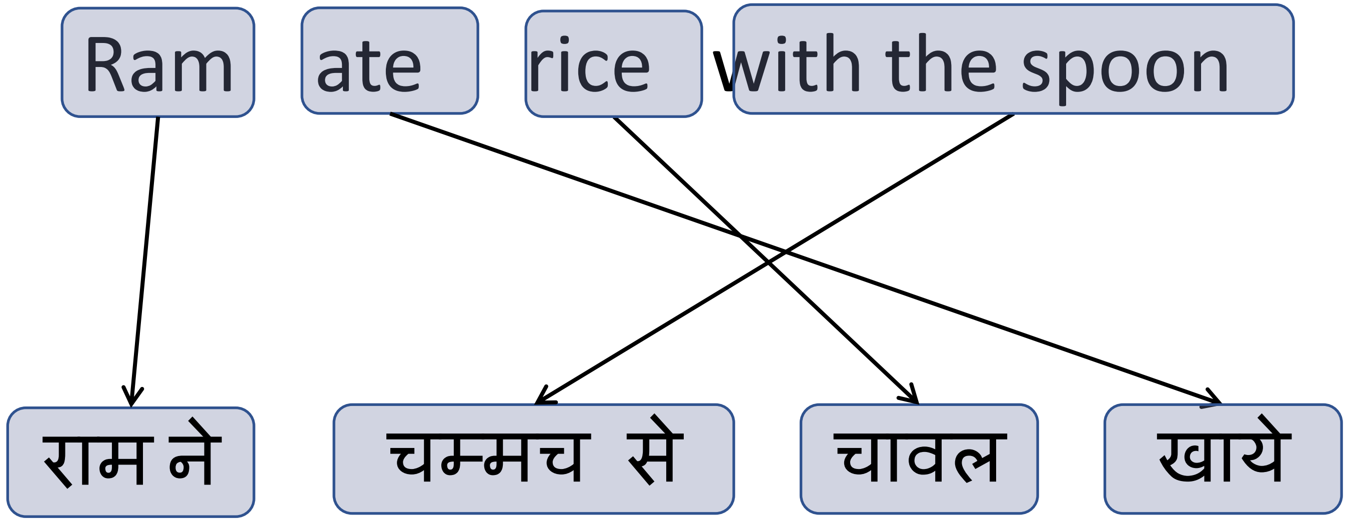
- *At each time decoder step:*
  - Feed model output from previous time step ➔ degrades performance
  - Feed ground-truth output from previous time step ➔ **teacher forcing**
- *Discrepancy in train and test scenarios* ➔ ***Exposure bias***
  - Solution → scheduled sampling
  - Sample from ground truth or predicted label
  - Sampling probability is varied: prefer ground truth earlier in training



Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent Neural networks. NeurIPS 2015.

Decoding

Searching for the best translations in the space of all translations



Decoding Strategies

- Exhaustive Search: *Score each and every possible translation – Forget it!* ➔ $O(V^N)$
- Sampling ➔ $O(NV)$
- Greedy ➔ $O(NV)$
- Beam Search ➔ $O(kNV)$

Greedy Decoding

w_1	0.03
w_2	0.7
w_3	0.05
w_3	0.1
w_4	0.08
w_5	0.04

Select best word using the distribution $P(y_j | y_{<j}, \boldsymbol{x})$

Sampling Decoding

w_1	0.03
w_2	0.7
w_3	0.05
w_3	0.1
w_4	0.08
w_5	0.04

Sample next word using the distribution $P(y_j | y_{<j}, \boldsymbol{x})$

Generate one word at a time sequentially

Not used to find best translation, but these methods have their uses ➔ for efficiency reasons

Greedy Search is not optimal

	t_1	t_2
w_1	**0.5**	0.1
w_2	0.4	0.2
w_3	0.05	**0.3**
w_3	0.02	0.1
w_4	0.01	0.1
w_5	0.02	0.2

Probability of best sequence w_1w_3 *=0.15*

	t_1	t_2
w_1	0.5	0.1
w_2	**0.4**	0.45
w_3	0.05	0.2
w_3	0.02	0.15
w_4	0.01	0.08
w_5	0.02	0.02

Probability of best sequence w_2w_2 *=0.18*

t_1 t_2

Beam Search

A compromise solution between greedy decoding and exhaustive search

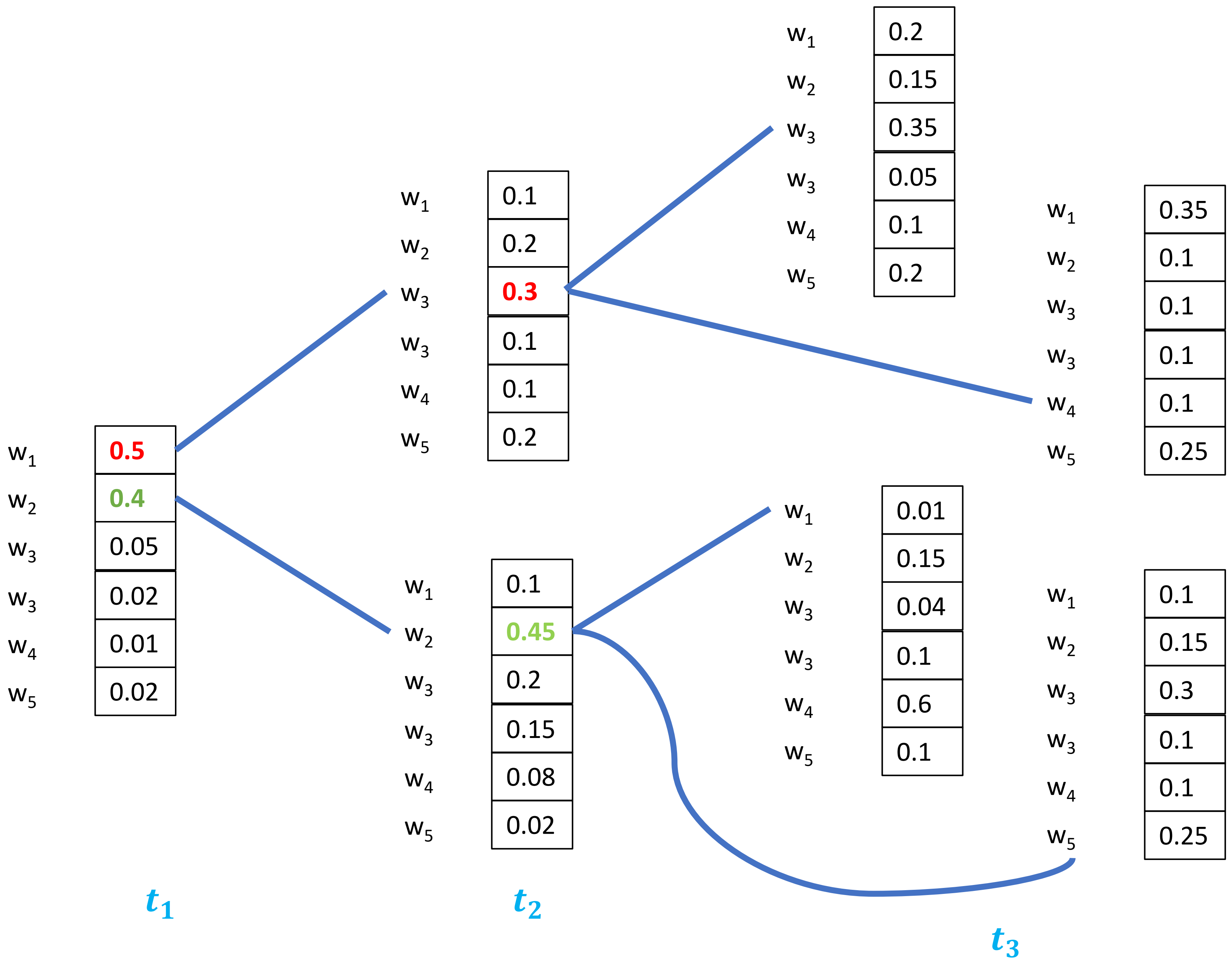
- *Explores more translation candidates than greedy search*
- *More efficient than exhaustive search*

2 Core Ideas:

- ***Incremental*** *construction & scoring of translation candidate (one decoder time step at a time)*
- *At each decoder time step,* ***keep the k-most probable partial translations***
    - *➔ these will be used for candidates expansion*

- ***Not guaranteed to find optimal solution ➔ search errors***

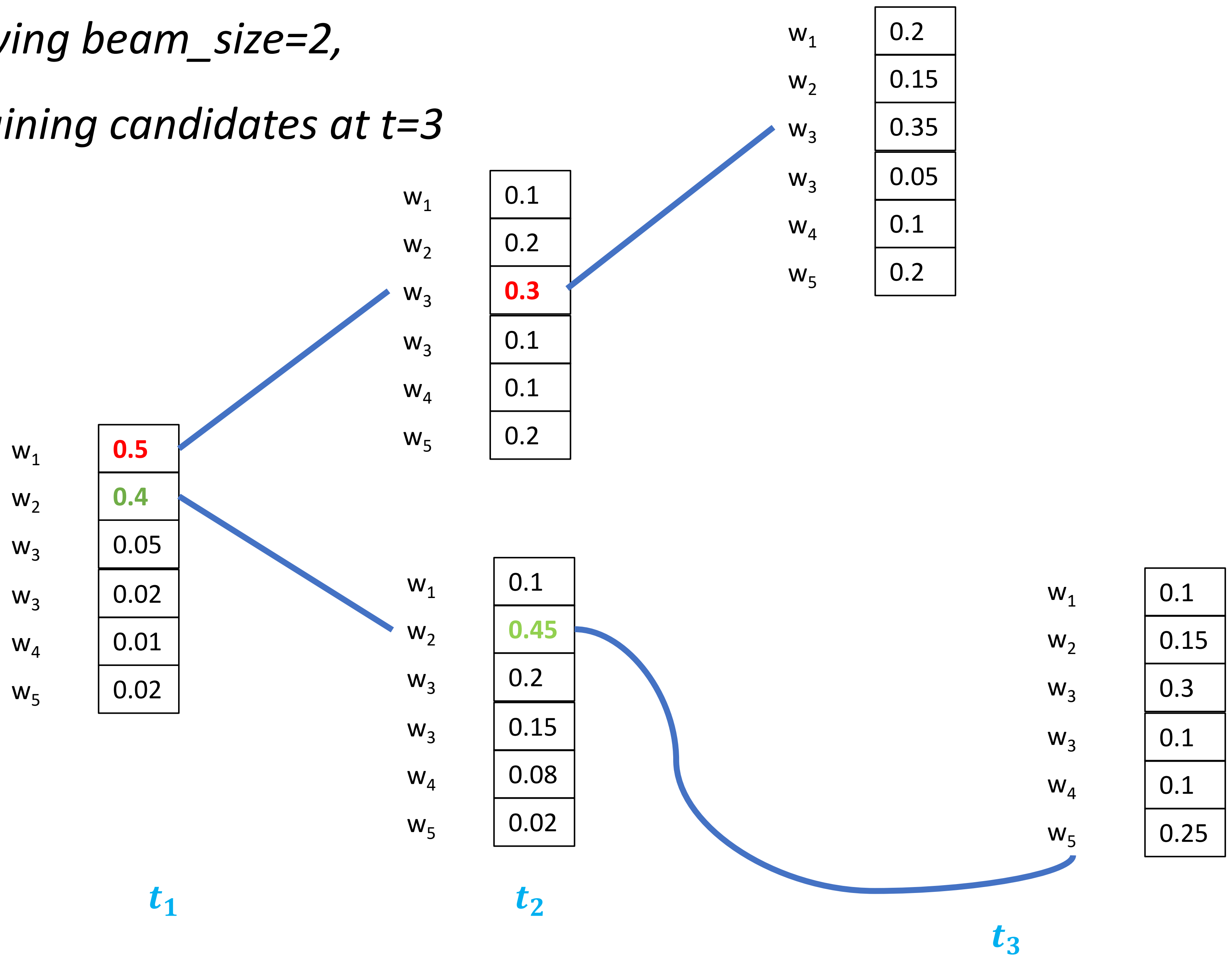
http://www.phontron.com/slides/nlp-programming-en-13-search.pdf



w1 0.5
w2 0.4
w3 0.05
w3 0.02
w4 0.01
w5 0.02
w1 0.1
w2 0.2
w3 0.3
w3 0.1
w4 0.1
w5 0.2
w1 0.1
w2 0.45
w3 0.2
w3 0.15
w4 0.08
w5 0.02
w1 0.2
w2 0.15
w3 0.35
w3 0.05
w4 0.1
w5 0.2
w1 0.35
w2 0.1
w3 0.1
w3 0.1
w4 0.1
w5 0.25
w1 0.01
w2 0.15
w3 0.04
w3 0.1
w4 0.6
w5 0.1
w1 0.1
w2 0.15
w3 0.3
w3 0.1
w4 0.1
w5 0.25
t1
t2
t3

Applying beam_size=2,

remaining candidates at t=3



Beam search tends to prefer shorter translations

Normalize the hypothesis score by the hypothesis length

$$S_{\text{LN}}(\mathbf{y}|\mathbf{x}) = \frac{\log P(\mathbf{y}|\mathbf{x})}{|\mathbf{y}|}$$

Or similar methods which offer tunable parameter (α) for the length penalty

$$S_{\text{LN-GNMT}}(\mathbf{y}|\mathbf{x}) = \log P(\mathbf{y}|\mathbf{x}) \frac{(1+5)^{\alpha}}{(1+|\mathbf{y}|)^{\alpha}}$$

Wu etal. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. https://arxiv.org/abs/1609.08144. 2016

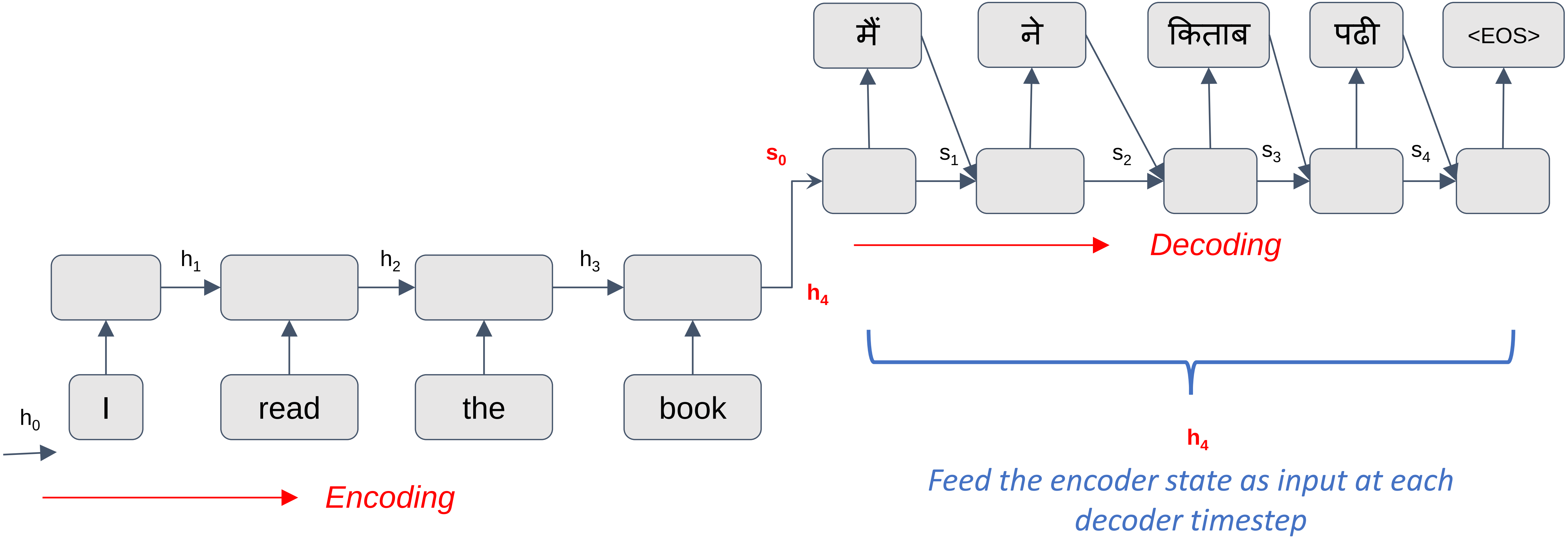
Topics

- *Why NMT?*
- *Encoder-Decoder Models*
- *Attention Mechanism*
- *Transformer Networks*
- *Backtranslation*
- *Subword-level Models*
- *Advanced Seq2Seq Modeling*

The entire source sentence is represented by a single vector

Problems

- Insufficient to represent to capture all the syntactic and semantic complexities
  - *Solution: Use a richer representation for the sentences*
- Long-term dependencies: Source sentence representation not useful after few decoder time steps
  - *Solution: Make source sentence information when making the next prediction*



मैं
ने
किताब
पढी
<EOS>
s0
s1
s2
s3
s4
Decoding
h1
h2
h3
h4
h0
I
read
the
book
Encoding
h4
Feed the encoder state as input at each decoder timestep

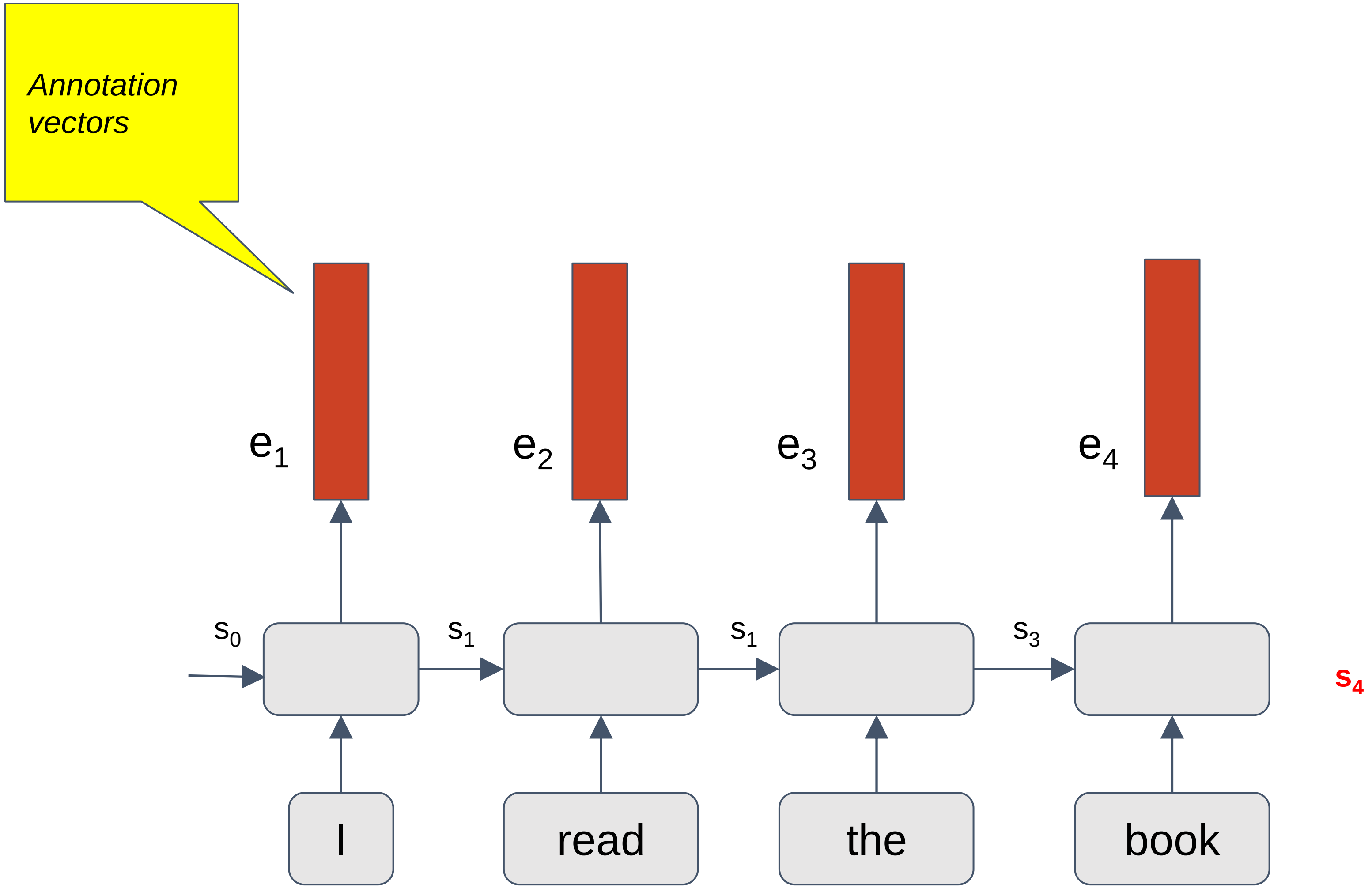
The entire source sentence is represented by a single vector

Problems

- Insufficient to represent to capture all the syntactic and semantic complexities
    - *Solution: Use a richer representation for the sentences*
- Long-term dependencies: Source sentence representation not useful after few decoder 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*

Let's call these encoder output vectors ***annotation vectors***

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *ICLR 2015.*

https://developer.nvidia.com/blog/introduction-neural-machine-translation-gpus-part-3/

How can the annotation vectors help predicting the next output?

Key Insight:

(1) Not all annotation vectors are equally important for prediction of the next element

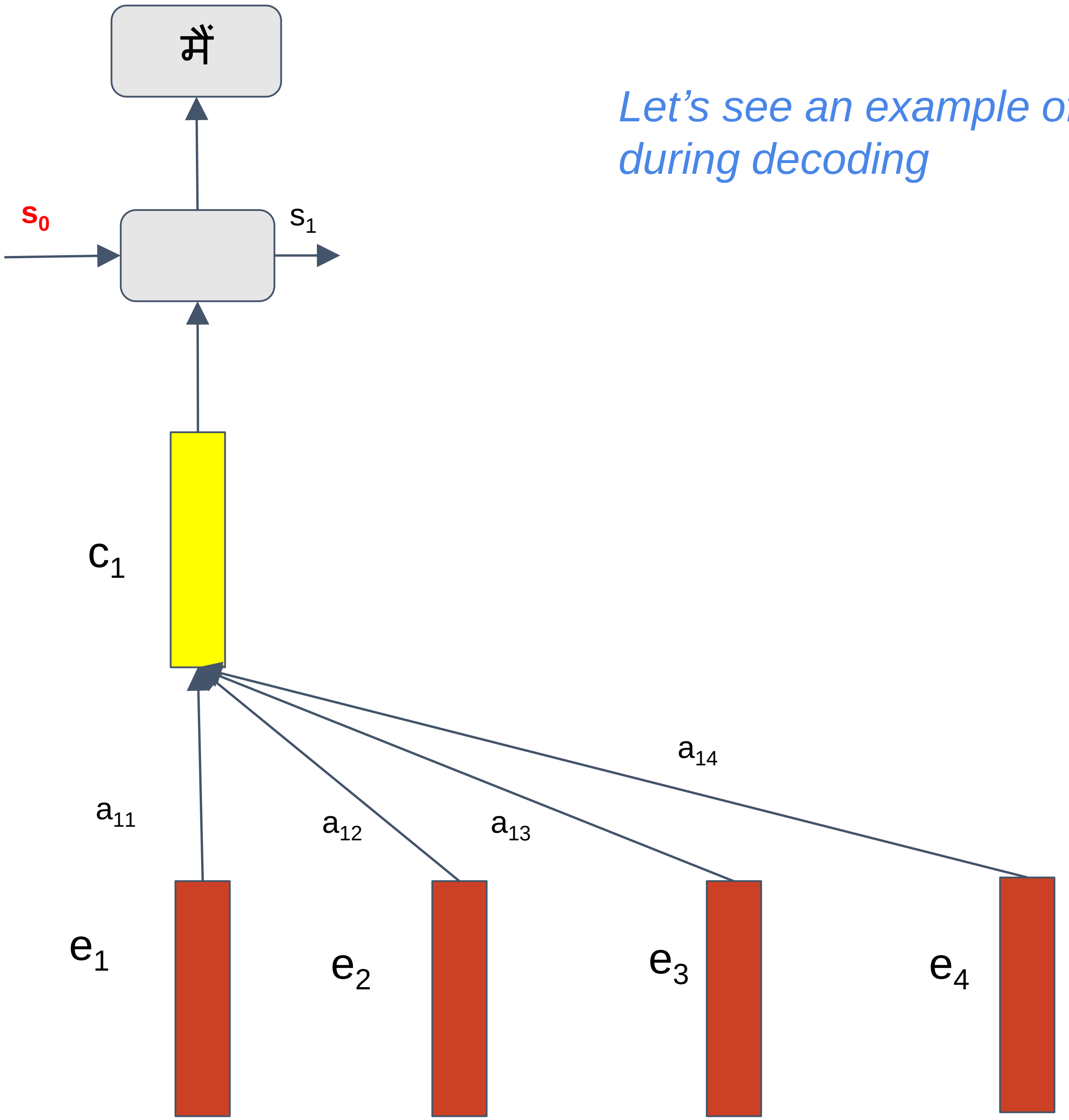
(2) The annotation vector to use next depends on what has been generated so far by the decoder

eg. To generate the 3rd target word, the 3rd source word is most important

Context vector = weighted average of the annotation vectors

More weight to annotation vectors which need more **focus or attention**

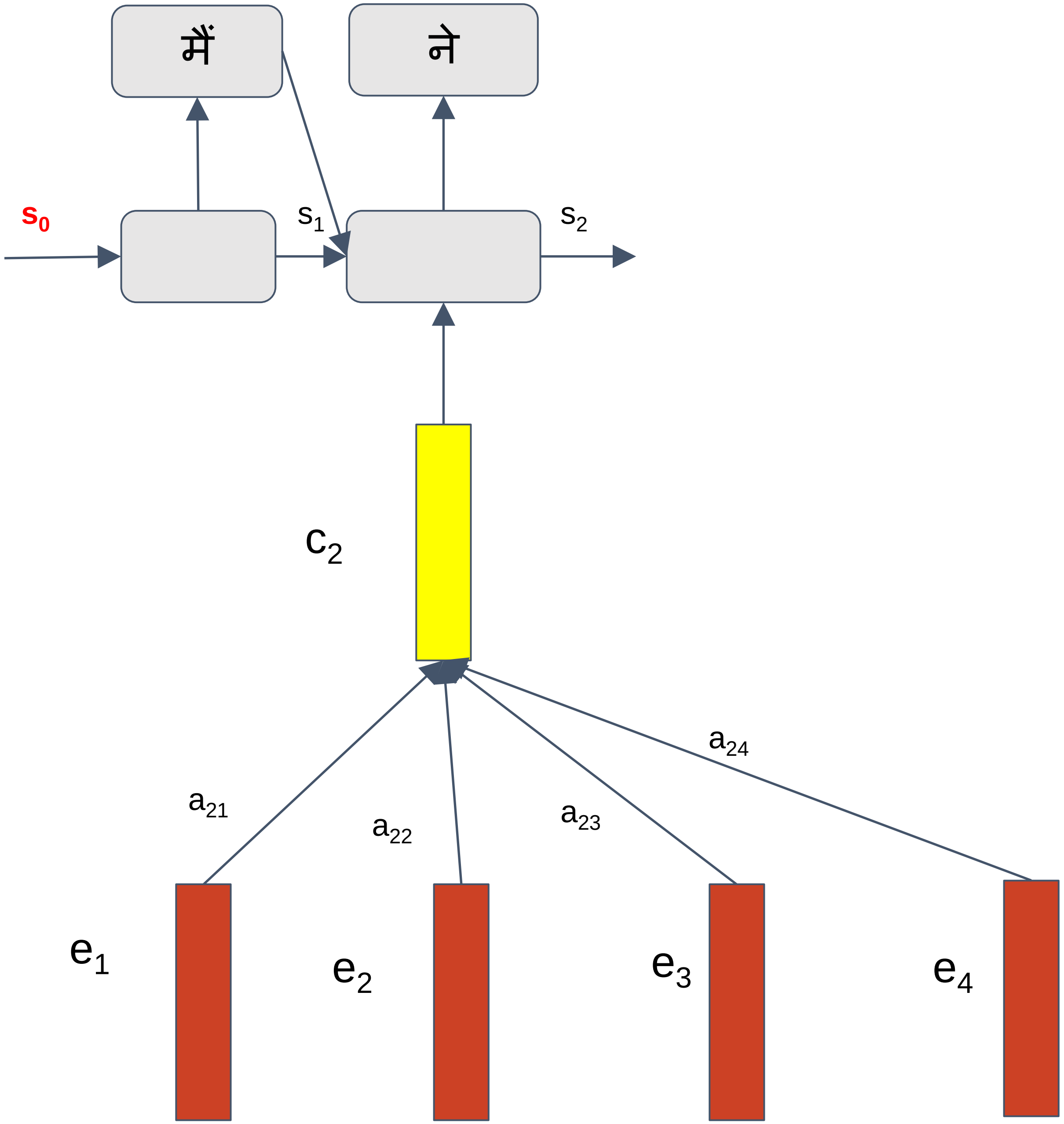
This averaged ***context vector*** is an input to the decoder

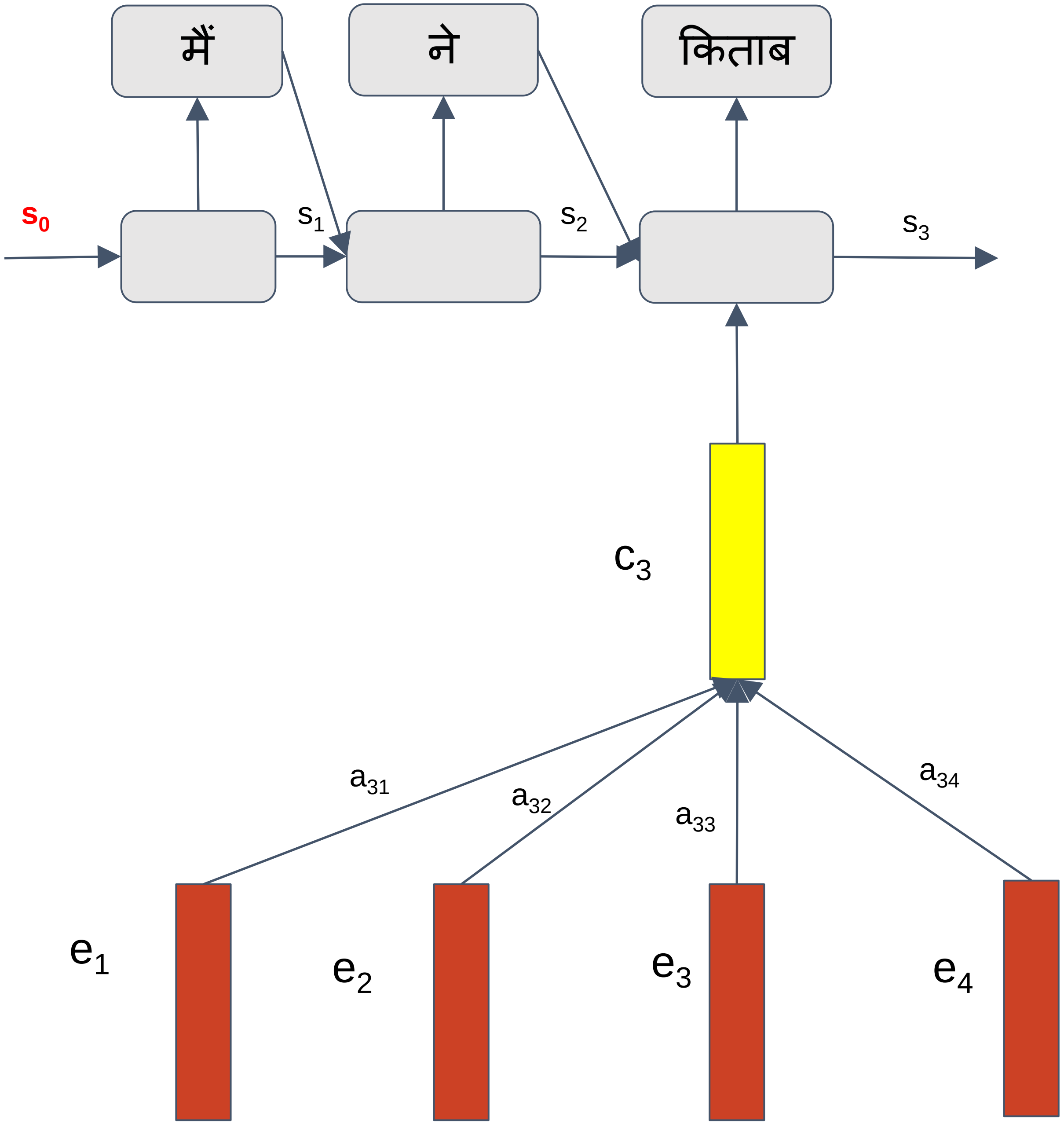


*Let's see an example of how the **attention mechanism** works during decoding*

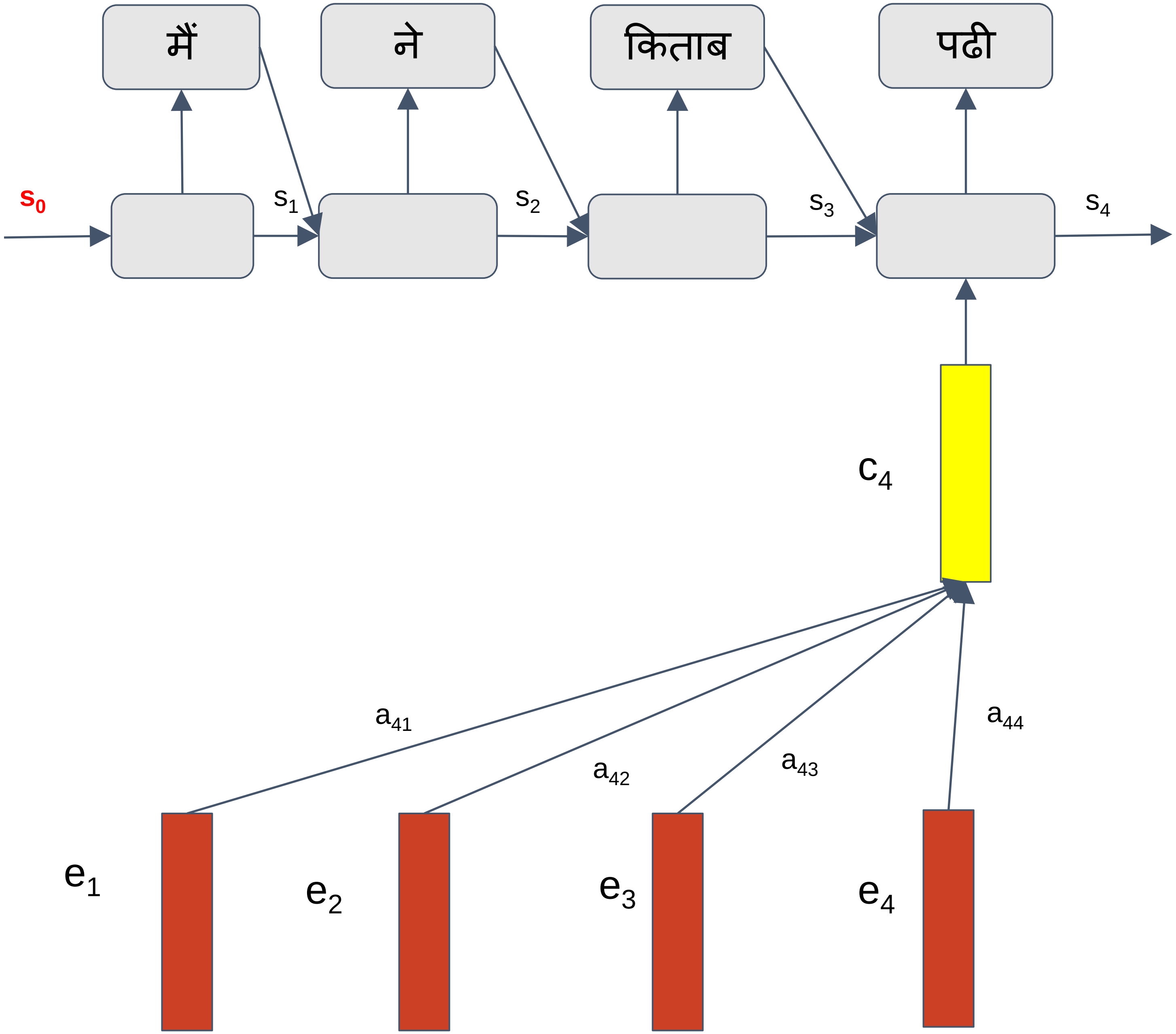
$$c_i = \sum_{j=1}^{n} a_{ij} e_j$$

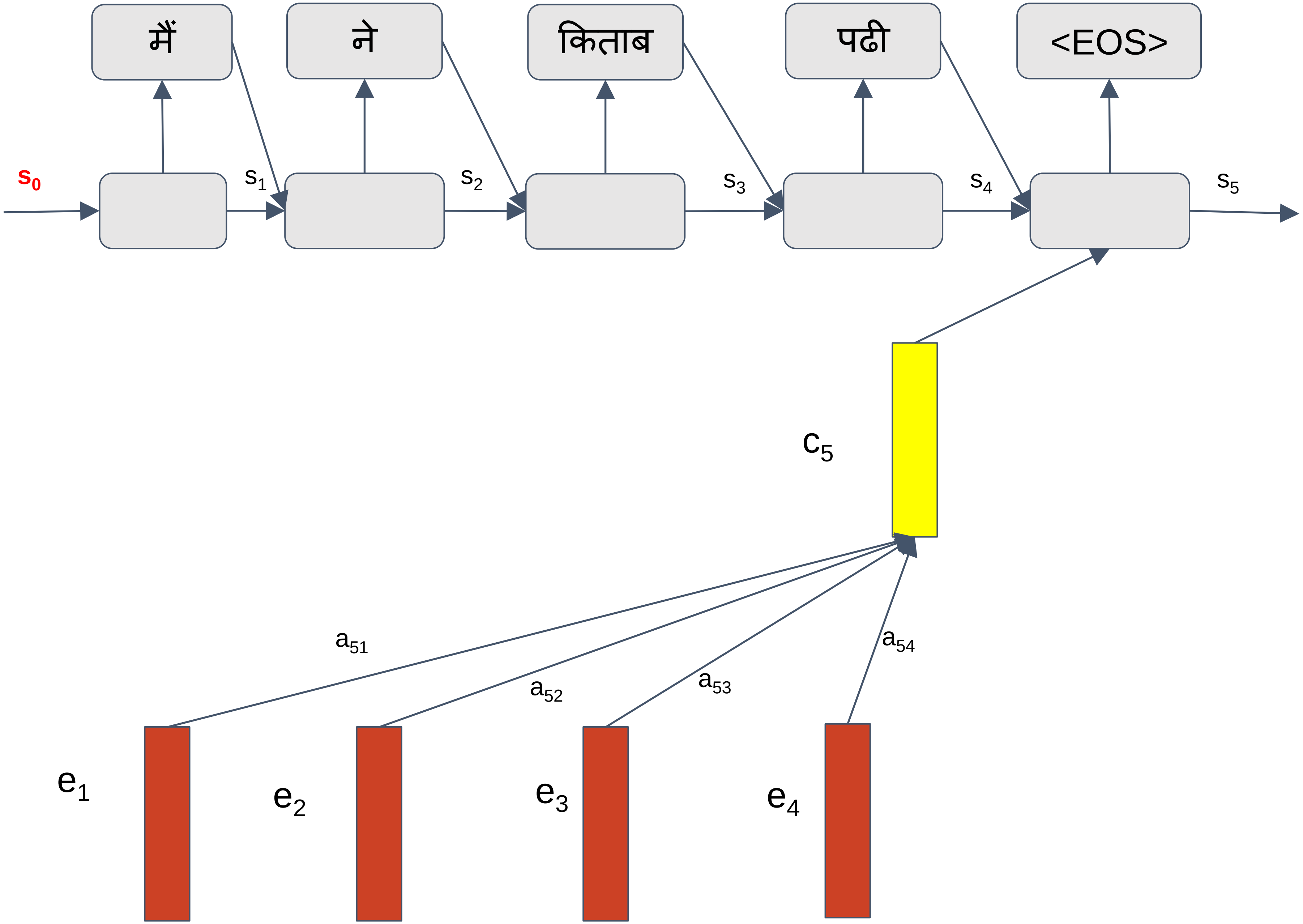
For generation of i^{th} output character:
c_i : context vector
a_{ij} : annotation weight for the j^{th} annotation vector
o_j: j^{th} annotation vector


मैं
ने
s0
s1
s2
c2
a21
a22
a23
a24
e1
e2
e3
e4



मैं
ने
किताब
s0
s1
s2
s3
c3
a31
a32
a33
a34
e1
e2
e3
e4


मैं
ने
किताब
पढी
s_0
s_1
s_2
s_3
s_4
c_4
a_{41}
a_{42}
a_{43}
a_{44}
e_1
e_2
e_3
e_4



मैं
ने
किताब
पढी
<EOS>
s_0
s_1
s_2
s_3
s_4
s_5
c_5
a_{51}
a_{52}
a_{53}
a_{54}
e_1
e_2
e_3
e_4

How do we find the attention weights?

Let the training data help you decide!!

Idea: Pick the attention weights that maximize the overall translation likelihood accuracy

Scoring function **g** to match the encoder and decoder states

$$\alpha_{ij} = g(s_{j-1}, e_i, \mathbf{emb}(y_{j-1}))$$

$$a_{ij} = \frac{\exp(\alpha_{ij})}{\sum_{i=1}^{i=N} \exp(\alpha_{kj})}$$

$$c_j = \sum_{i=1}^{i=N} a_{ij} e_i$$

How do we find the attention weights?

Let the training data help you decide!!

Idea: Pick the attention weights that maximize the overall translation likelihood accuracy

$$\alpha_{ij} = g(s_{j-1}, e_i, \mathbf{emb}(y_{j-1}))$$

g can be a feedforward network or a similarity metric like dot product

$$a_{ij} = \frac{\exp(\alpha_{ij})}{\sum_{i=1}^{i=N} \exp(\alpha_{kj})}$$

$$c_j = \sum_{i=1}^{i=N} a_{ij} e_i$$

How do we find the attention weights?

Let the training data help you decide!!

Idea: Pick the attention weights that maximize the overall translation likelihood accuracy

$$\alpha_{ij} = g(s_{j-1}, e_i, \mathbf{emb}(y_{j-1}))$$

Normalize score to obtain attention weights

$$a_{ij} = \frac{\exp(\alpha_{ij})}{\sum_{i=1}^{i=N} \exp(\alpha_{kj})}$$

$$c_j = \sum_{i=1}^{i=N} a_{ij} e_i$$

How do we find the attention weights?

Let the training data help you decide!!

Idea: Pick the attention weights that maximize the overall translation likelihood accuracy

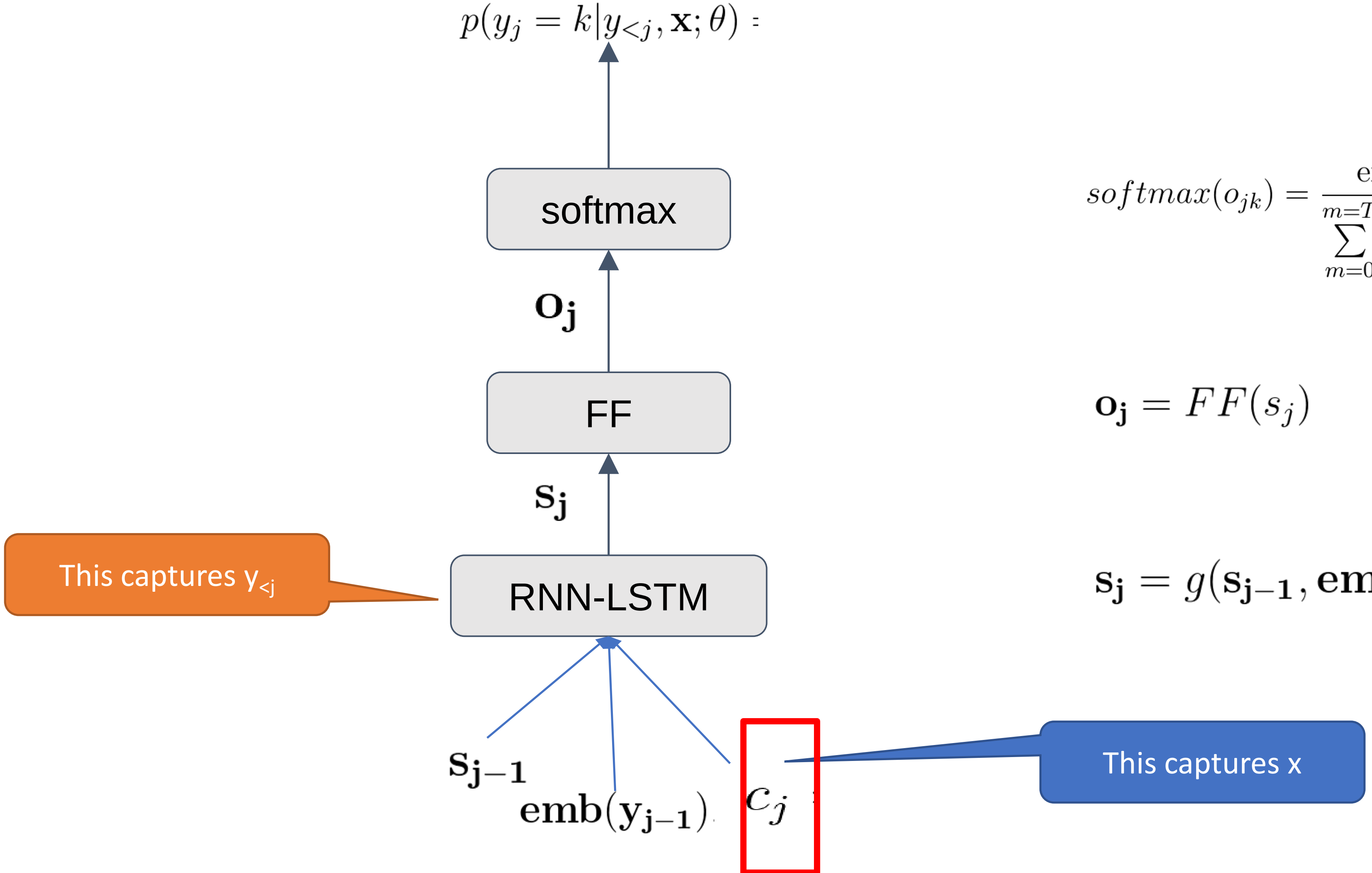
$$\alpha_{ij} = g(s_{j-1}, e_i, \mathbf{emb}(y_{j-1}))$$

$$a_{ij} = \frac{\exp(\alpha_{ij})}{\sum_{i=1}^{i=N} \exp(\alpha_{kj})}$$

Final context vector is weighted average of encoder outputs

$$c_j = \sum_{i=1}^{i=N} a_{ij} e_i$$

Let us revisit what the decoder does at time step t



$$softmax(o_{jk}) = \frac{\exp(o_{jk})}{\sum_{m=0}^{m=T} \exp(o_{jm})}$$

$$\mathbf{o_j} = FF(s_j)$$

$$\mathbf{s_j} = g(\mathbf{s_{j-1}}, \mathbf{emb(y_{j-1})}, \mathbf{c})$$

Choice of Attention Scoring Function

Feedforward : $\alpha_{ij} = \boldsymbol{W_a}[e_j; s_i]$

Dot Product : $\alpha_{ij} = s_i^T e_j$

Scaled Dot Product : $\alpha_{ij} = \frac{s_i^T e_j}{\sqrt{|e_j|}}$

Multiplicative Attention : $\alpha_{ij} = s_i^T \boldsymbol{W_a} e_j$

Additive Attention : $\alpha_{ij} = \boldsymbol{W_1} s_i + \boldsymbol{W_2} e_j$

Effective Approaches to Attention-based Neural Machine Translation. Thang Luong, Hieu Pham, Christopher D. Manning. EMNLP 2015.

Attention is a general and important concept in Deep learning

Given a set of VALUES ➔ select a summary of the values that is relevant to a QUERY

Each VALUE represented by a KEY ➔ the QUERY is matched to the KEY (content similarity)

Select a summary with different focus on different values ➔ Weighted average

Associative memory read + selection

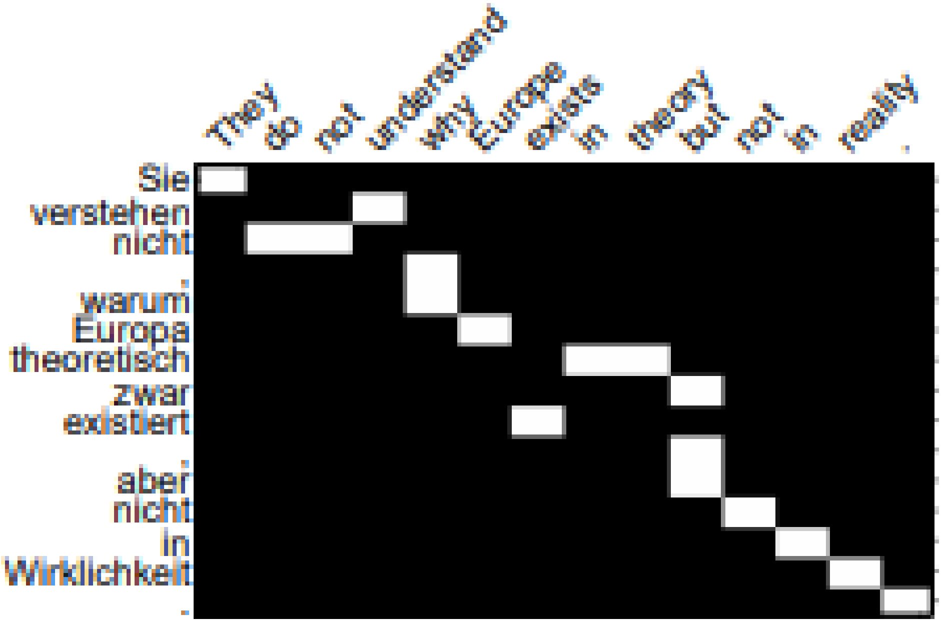
<u>For MT</u>

QUERY: decoder state

VALUE, KEY: encoder annotation vector

Benefits of Attention

- Significant improves in NMT quality
  - Performs better on long sentences
  - Word-order is no longer a major issue for
  - Used in all NMT systems
- Attention provides some interpretability
  - Attention!=Alignment
- There is more to attention



https://arxiv.org/pdf/1508.04025.pdf

Benefits of Neural MT

~~Limitations of SMT~~

- *No end-to-end optimization* ➔ *Single optimization objective that accounts for alignment, word reordering*
- *Divergent word-order is a big challenge* ➔ *Attention mechanism*
- *n-gram LM not the best way to score translation fluency*

  ➔ *Target conditional-RNN LM (no standalone LM)*
- *Model size is a function of the data size*

  ➔ *Give architecture, model size is fixed*

Topics

- *Why NMT?*
- *Encoder-Decoder Models*
- *Attention Mechanism*
- *Transformer Networks*
- *Backtranslation*
- *Subword-level Models*
- *Advanced Seq2Seq Modeling*

Transformer Architecture

Limitations of Recurrent Architectures

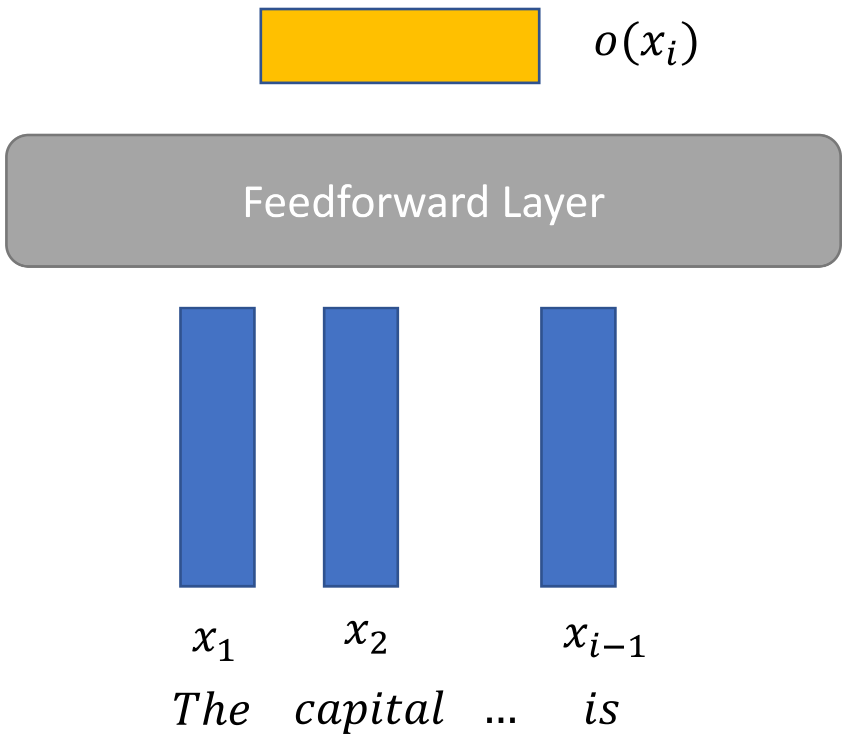
- Elements of a sequence have to processed serially
  - Pro: Number of computations linear in the length of the sequence
  - Con: Encoding time $\propto$ Length of sequence
- Not effective at modeling long-term dependencies

Revisiting idea → Compare every elements with all other elements

We have already seen this …

Feedforward Network can handle only a limited context

Can we approach this problem differently?

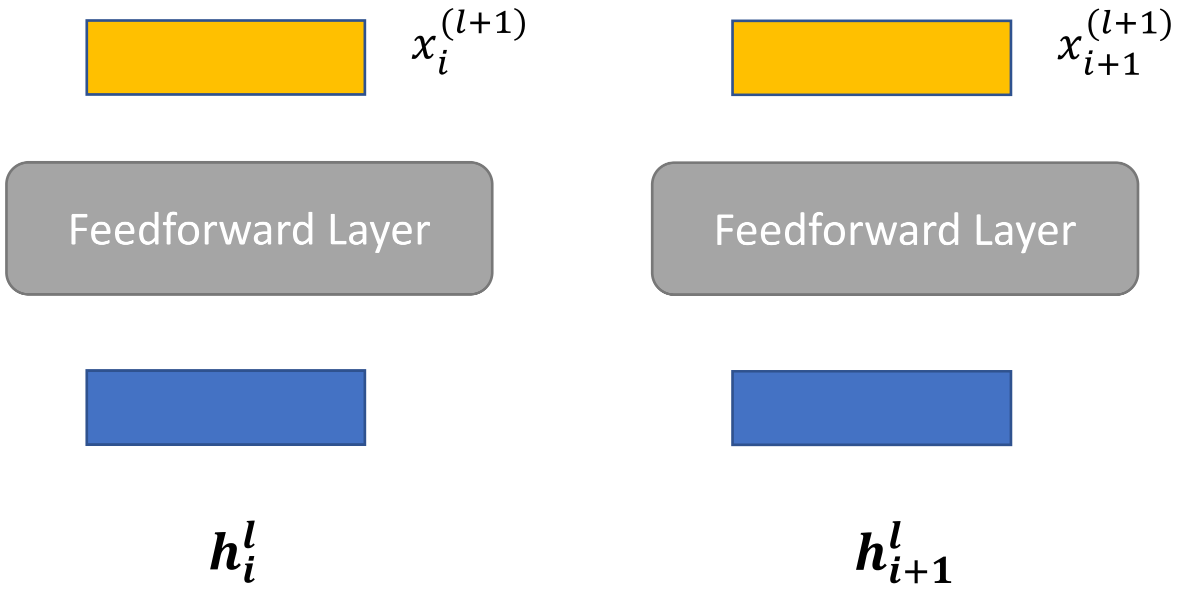


Revisiting idea ➔ Compare every elements with all other elements

Represent the input context as a weighted average of input word embeddings

$$\boldsymbol{h}_i^l = \sum_{i=1}^{N} \boldsymbol{w}_i x_i$$

$$\boldsymbol{x}_i^{l+1} = \boldsymbol{FF}(h_i^l)$$



Non-recurrent ➔ this operation can be applied in parallel to all elements in the sequence

How do we compute weights ➔ Attention!

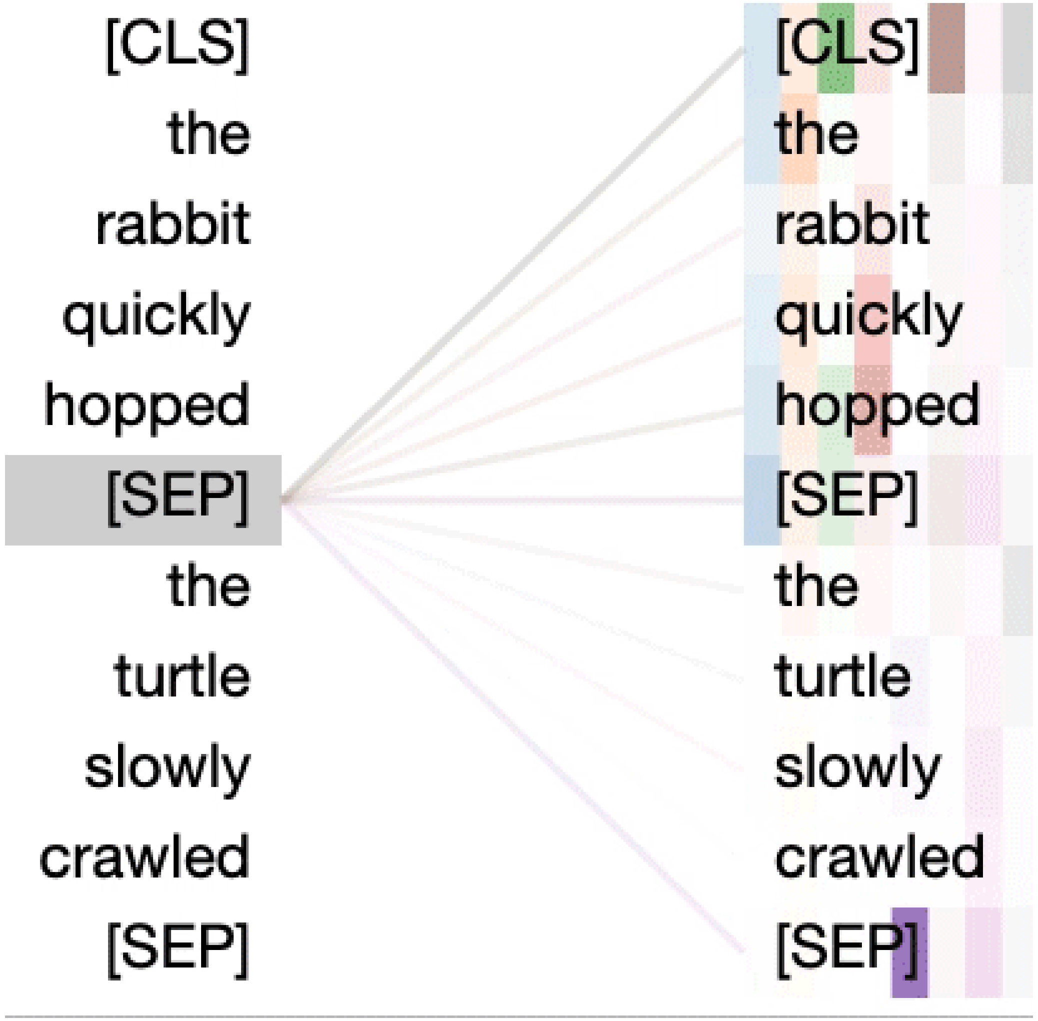
Self-Attention

Every word is compared to every other word in the same sentence

x_i ➔ query

$x_1, x_2\ x_3 \dots x_n$ ➔ values

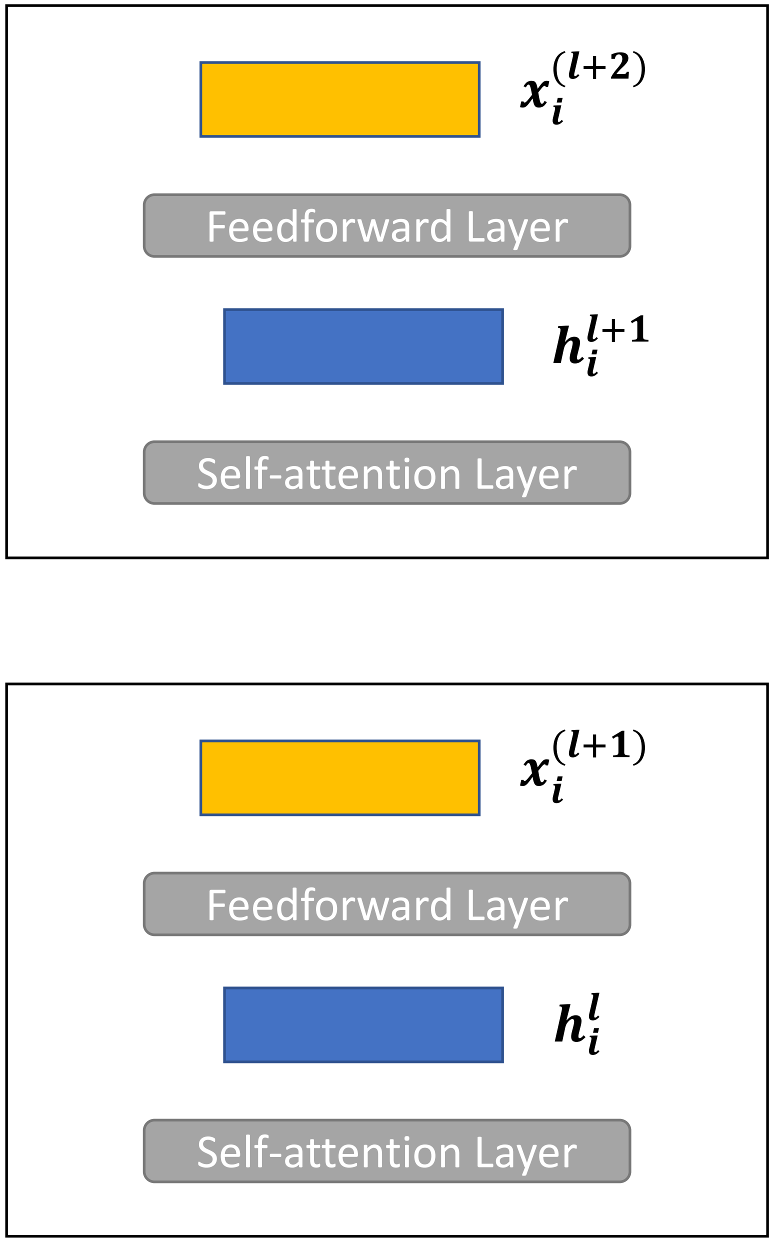
Direct comparison between arbitrary words ➔
long-range dependencies can be better modelled



More computations than Recurrent models: $O(n^2)$

Transformer Architecture

Stack self-attention blocks to create deep networks



Positional Embeddings

The ICICI bank branch is the bank of the river

The self-attention model has no notion of position,

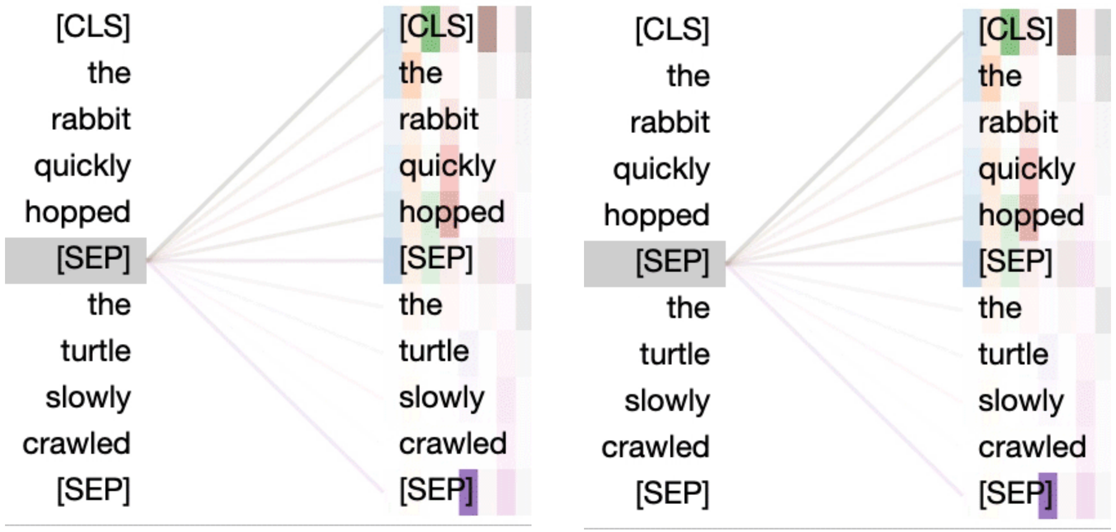
➔ same words will have same representations irrespective of their position/syntactic role in the sentence

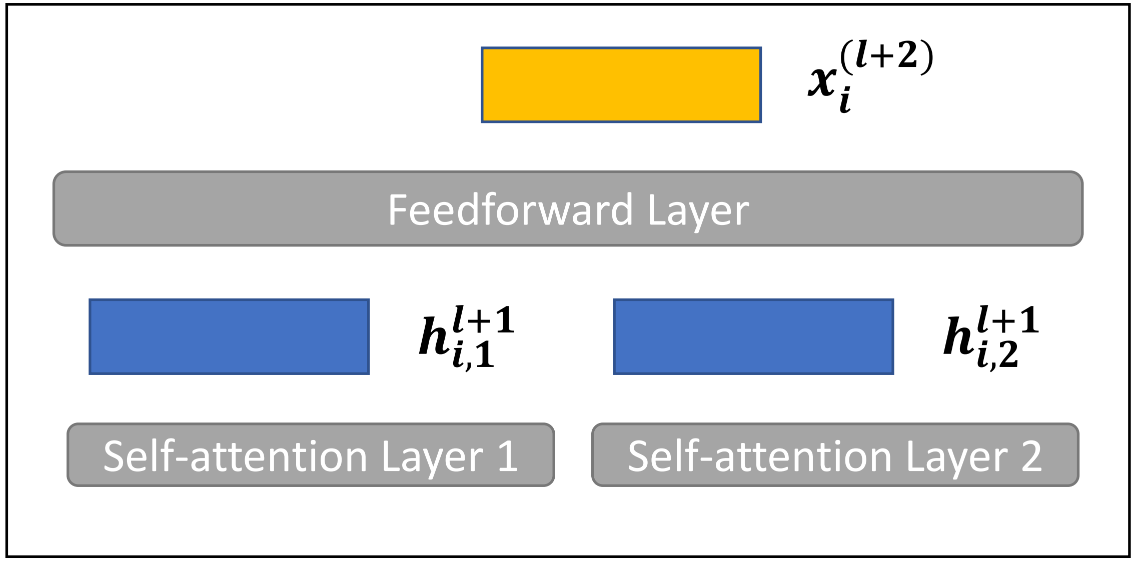
Create positional embeddings that uniquely and deterministically identify a position

Add it to the word embedding at the bottom layer

https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

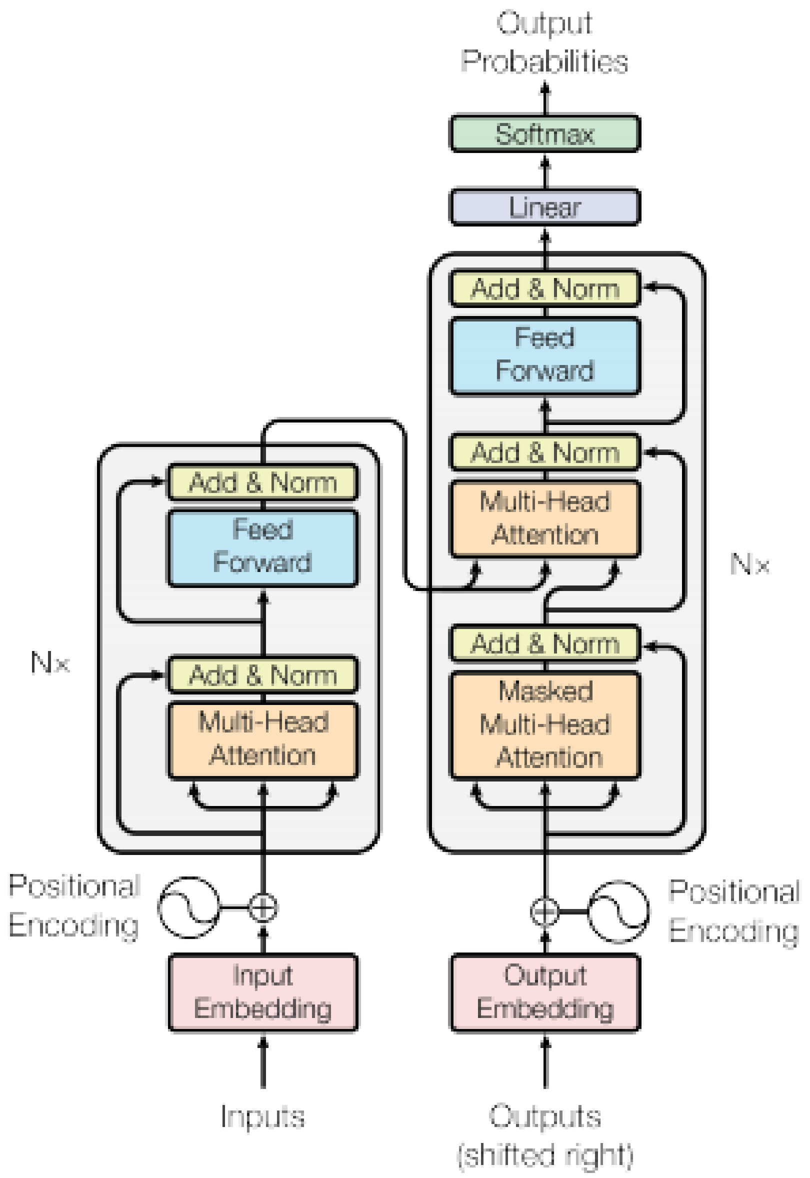
Multiple self-attention heads





Multiple self-attention networks at each layer

Each head learns different kinds of dependencies



Putting it all together

Decoder layer also has a cross-attention layer

Decoder ➔ masking for future time-steps while computing self-attention

There are residual connections & layer-normalization between layers

http://nlp.seas.harvard.edu/2018/04/03/attention.html
http://jalammar.github.io/illustrated-transformer/

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." NeurIPS (2017).

Transformer has led to tremendous advances in MT

Encoder architectures like BERT based on Transformer have yielded large improvements in NLU tasks

Transformer models are the de-facto standard models for many NLP tasks

Topics

- *Why NMT?*
- *Encoder-Decoder Models*
- *Attention Mechanism*
- *Transformer Networks*
- *Backtranslation*
- *Subword-level Models*
- *Advanced Seq2Seq Modeling*

The models discussed so far do not use monolingual data

Can monolingual data help improve NMT models?

Backtranslation

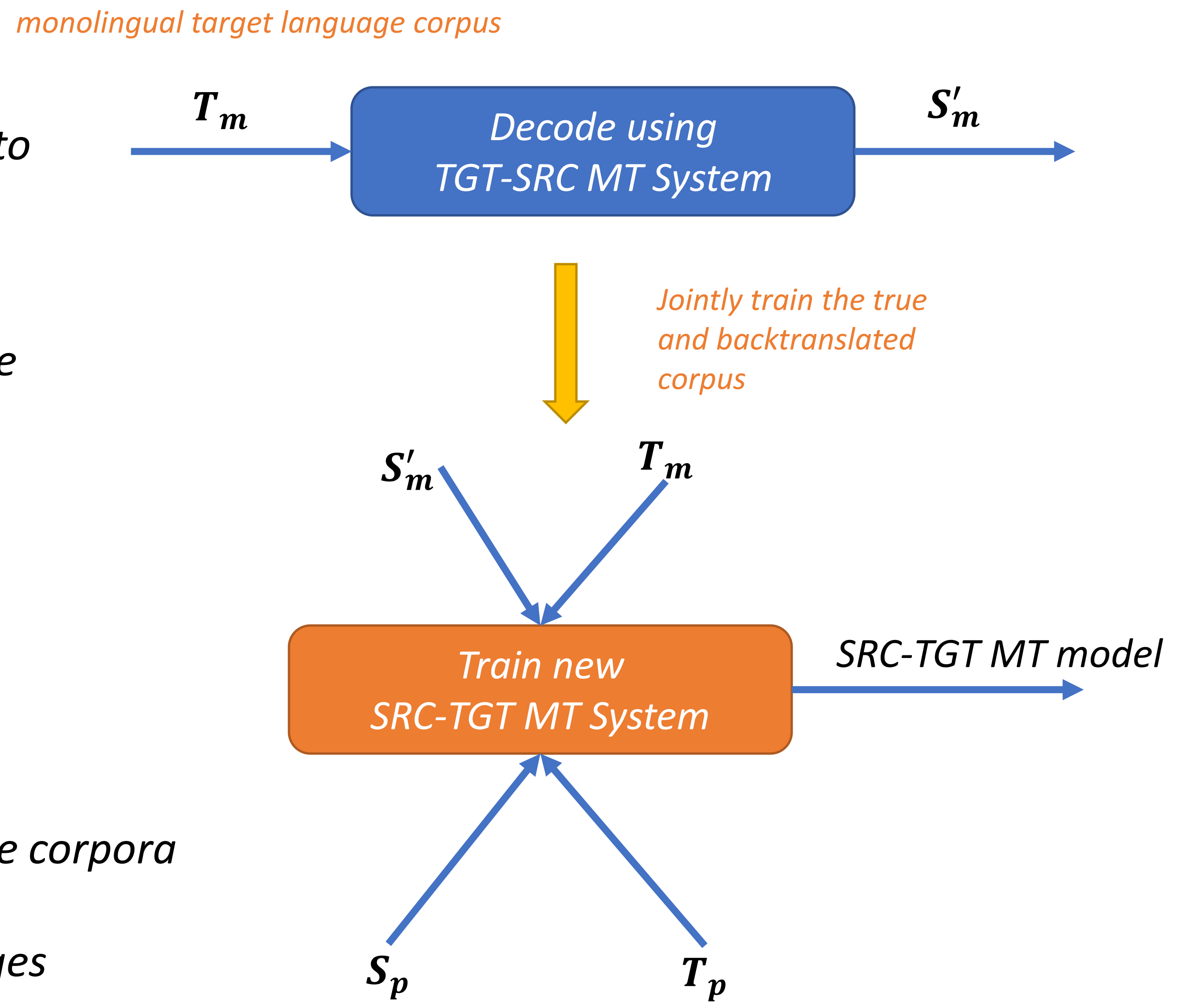
Create pseudo-parallel corpus using Target to source model (Backtranslated corpus)

Need to find the right balance between true and backtranslated corpus

Why is backtranslation useful?

- *Target side language model improves*
  - *target side is clean*
- *Adaptation to target language domain*
- *Prevent overfitting by exposure to diverse corpora*

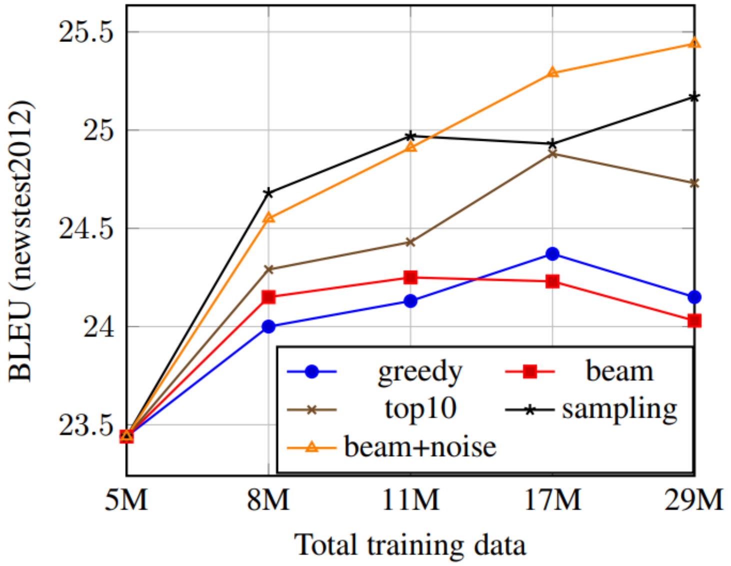
Particularly useful for low-resource languages



Sennrich, Rico, Barry Haddow, and Alexandra Birch. "Improving neural machine translation models with monolingual data." ACL 2016

Make backtranslation more diverse

- *Sampling*
- *Restricted Sampling*
- *Beam+noising*



Make it easy for the model to distinguish between natural & synthetic input

Tagged Backtranslation ➔

add a special token indicating that the input is synthetic

Noise type	Example sentence
[no noise]	Raise the child, love the child.
P3BT	child Raise the, love child the.
NoisedBT	Raise child ___ love child, the.
TaggedBT	<BT> Raise the child, love the child.
TaggedNoisedBT	<BT> Raise, the child the ___ love.

Tagged BT and Noised BT serve the same purpose ➔ distinguishing inputs

Sergey Edunov, Myle Ott, Michael Auli, David Grangier . Understanding Back-Translation at Scale. EMNLP 2018.
Isaac Caswell, Ciprian Chelba, David Grangier. Tagged Back-Translation. WMT 2019

Self Training

Create pseudo-parallel corpus using initial source to target model (Forward translated corpus)

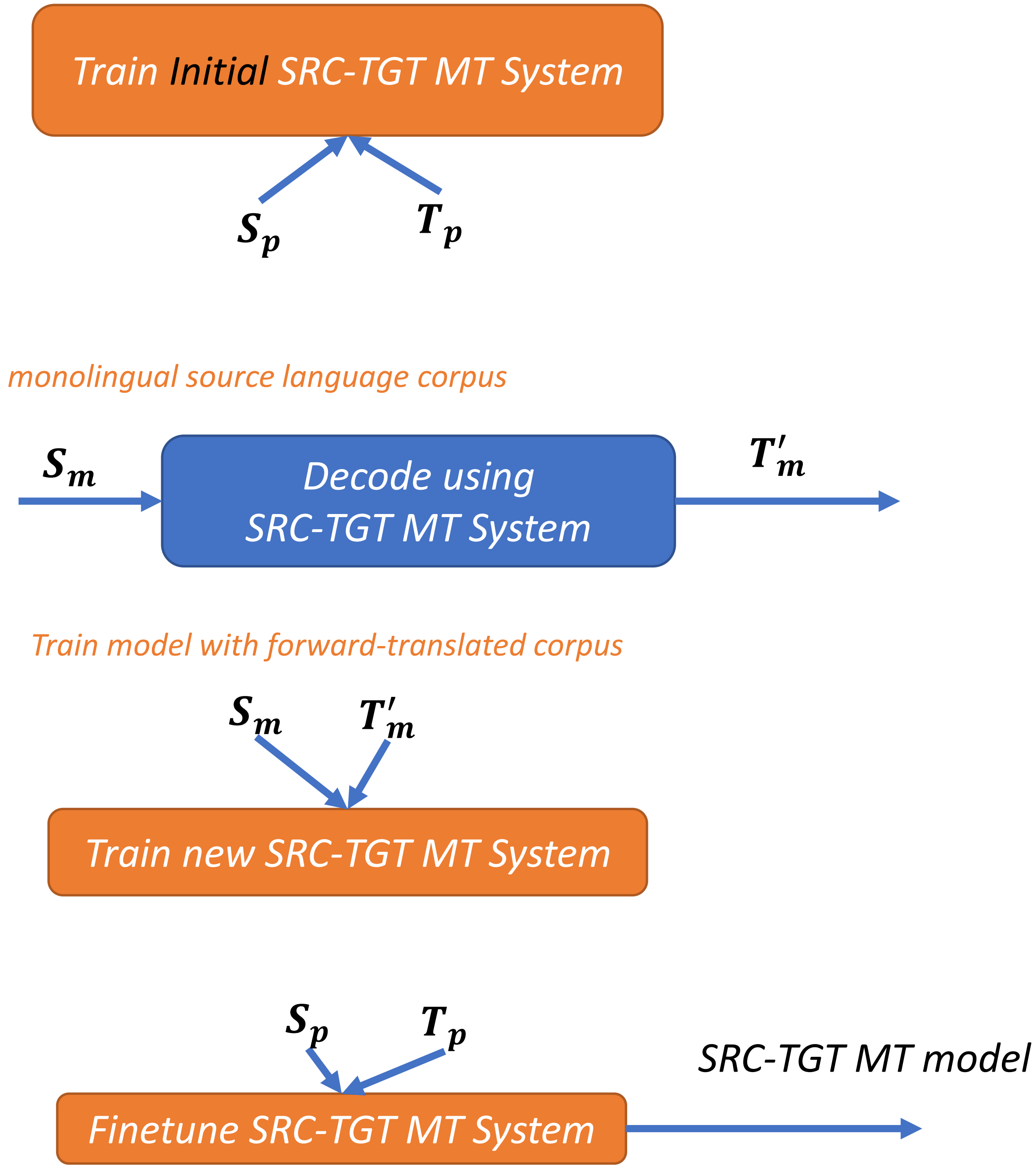
Target side of pseudo-parallel corpus is noisy

- *Train the S-T mode on pseudo-parallel corpora*
- *Tune on true parallel corpora*
- *(Noising the input helps, use beam search)*

Why is self-training useful?

- *Noise plays an important role*
- *Adaptation to source language domain*
- *Prevent overfitting by exposure to diverse corpora*

Works well if the initial model is reasonably good



Junxian He, Jiatao Gu, Jiajun Shen, Marc'Aurelio Ranzato. Revisiting Self-Training for Neural Sequence Generation. ICLR 2020.

Topics

- *Why NMT?*
- *Encoder-Decoder Models*
- *Attention Mechanism*
- *Transformer Networks*
- *Backtranslation*
- *Subword-level Models*
- *Advanced Seq2Seq Modeling*

The Vocabulary Problem

- **The input & output embedding layers are finite**
  - How to handle an open vocabulary?
  - How to translate named entities?

- **Softmax computation at the output layer is expensive**
  - Proportional to the vocabulary size

$$softmax(o_{jk}) = \frac{\exp(o_{jk})}{\sum\limits_{m=0}^{m=T} \exp(o_{jm})}$$

Subword-level Translation

Original sentence: प्रयागराज में 43 दिनों तक चलने वाला माघ मेला आज से शुरू हो गया है

Possible inputs to NMT system:

- प्रयाग @@राज में 43 दि @@नों तक चल @@ने वाला माघ मेला आज से शुरू हो गया है
- प्र या ग रा ज _में _43 _दि नों _त क _च ल ने _ वा ला _मा घ मे ला _आज _से _शुरू _हो _गया _है

Obvious Choices: Character, Character n-gram, Morphemes ➔ They all have their flaws!

The New Subword Representations: Byte-Pair Encoding, Unigram (implemented in SentencePiece package)

Learn a fixed vocabulary & segmentation model from training data

↓

Segment Training Data based on vocabulary

↓

Train NMT system on the segmented model

{प्रयाग, राज, में दि, नों, तक, चल, ने}

vocabulary

{प्रयाग राज}
{च ल}
{चल, ने}

Segmentation model

प्रयाग@@राज में 43 दि@@नों तक चल@@ने वाला माघ मेला आज से शुरू हो गया है

- Every word can be expressed as a concatenation of subwords
- A small subword vocabulary has good representative power
    - 4k to 64k depending on the size of the parallel corpus
- Most frequent words should not be segmented

Byte Pair Encoding

Byte Pair Encoding is a greedy compression technique (Gage, 1994)

Number of BPE merge operations=3
Vocab: A B C D E F

P_1=AD P_2=EE P_3=P_1D

Words to encode *Iterations*

Words to encode	1	2	3	4
BADD	BADD	BP_1D	BP_1D	BP_3
FAD	FAD	FP_1	FP_1	FP_1
FEEDE	FEEDE	FEEDE	FP_2DE	FP_2DE
ADDEEF	ADDEEF	P_1DEEF	P_1DP_2F	P_3P_2F

Data-dependent segmentation

- Inspired from compression theory
- MDL Principle *(Rissansen, 1978)* ⇒ Select segmentation which maximizes data likelihood

Problems with subword level translation

Unwanted splits:

नाराज़ → ना राज़ ➔ no secret

Problem is exacerbated for:

- Named Entities
- Rare Words
- Numbers

Explore multiple subword segmentations

- BPE dropout
- Unigram + subword-regularization

Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates. Taku Kudo. ACL 2018.
BPE-Dropout: Simple and Effective Subword Regularization Ivan Provilkov, Dmitrii Emelianenko, Elena Voita. ACL 2020

Topics

- *Why NMT?*
- *Encoder-Decoder Models*
- *Attention Mechanism*
- *Transformer Networks*
- *Backtranslation*
- *Subword-level Models*
- *Advanced Seq2Seq Modeling*

Advanced S2S Modeling

Benefits of Neural MT

Opens-up new possibilities

- *Multi-source translation models*
- *Transfer Learning*
- *Cross-lingual/Multilingual MNMT*
- *Unsupervised NMT*
- *Multimodal translation*
- *End-to-End Speech-to-Speech Translation*
- *Document-level Translation*

Learning Word Alignments

Attention!=Alignment ➔ Can we get good alignments with supervision?

Word-alignments from statistical aligner ➔ train NMT with additional objectives

$$\mathcal{L}_\theta = \sum_{(\mathbf{x},\mathbf{y},\hat{\alpha})\in\mathbf{C}} -\log p(\mathbf{y}|\mathbf{x};\theta) + \lambda \times \Delta(\alpha,\hat{\alpha})$$

Δ measures the distance between true alignment ($\hat{\alpha}$) and attention weights (α)

Δ can be modelled using least square error, cross entropy, etc.

1. Liu, L., Utiyama, M., Finch, A., & Sumita, E. Neural machine translation with supervised attention. *COLING 2016*.
2. Zenkel, T., Wuebker, J., & DeNero, J. End-to-end neural word alignment outperforms GIZA++. *ACL 2020.*

Sequence level training objectives

Problems with Maximum Likelihood Estimation

- ***Exposure bias**: Models are exposed to training data, not model predictions during training*
- ***Word-level objective** that does not correspond to MT quality metrics*

Solution: Directly optimize evaluation metrics with model predicted outputs

Evaluation metrics: BLEU, TER, etc. … more on that later

Maximum Likelihood Estimation

$$\hat{\boldsymbol{\theta}}_{\text{MLE}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left\{ \mathcal{L}(\boldsymbol{\theta}) \right\},$$

where

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_{s=1}^{S} \log P(\mathbf{y}^{(s)} | \mathbf{x}^{(s)}; \boldsymbol{\theta})$$
$$= \sum_{s=1}^{S} \sum_{n=1}^{N^{(s)}} \log P(\mathbf{y}_n^{(s)} | \mathbf{x}^{(s)}, \mathbf{y}_{<n}^{(s)}; \boldsymbol{\theta}).$$

Minimum Risk Training

$$\hat{\boldsymbol{\theta}}_{\text{MRT}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \left\{ \mathcal{R}(\boldsymbol{\theta}) \right\}.$$

$$\mathcal{R}(\boldsymbol{\theta}) = \sum_{s=1}^{S} \mathbb{E}_{\mathbf{y}|\mathbf{x}^{(s)};\boldsymbol{\theta}} \left[\Delta(\mathbf{y}, \mathbf{y}^{(s)}) \right]$$
$$= \sum_{s=1}^{S} \sum_{\mathbf{y} \in \mathcal{Y}(\mathbf{x}^{(s)})} P(\mathbf{y} | \mathbf{x}^{(s)}; \boldsymbol{\theta}) \Delta(\mathbf{y}, \mathbf{y}^{(s)})$$

Difficult to enumerate all translations

Sample Translations

Shen et al. Minimum Risk Training for Neural Machine Translation. ACL 2016.

Modeling Coverage in Translation

Attention weights Computation ignores past attention weights

$$\alpha_{ij} = g(s_{j-1}, e_i, \mathbf{emb}(y_{j-1}))$$

Under-translation and over-translation

Coverage vector to keep track if source has been translated

$$\mathbb{C}_{i,j} = f(s_{j-1}, e_i, \alpha_{ij}, \mathbb{C}_{i,j-1})$$

Coverage vector also updated in end-to-end training

$$\alpha_{ij} = g(s_{j-1}, e_i, \mathbf{emb}(y_{j-1}), \mathbb{C}_{i,j})$$

Tu, Zhaopeng, et al. *Modeling coverage for neural machine translation*. *ACL 2016.*

Pointer Generator Networks

Lionel Messi won his first internal trophy for Argentina.

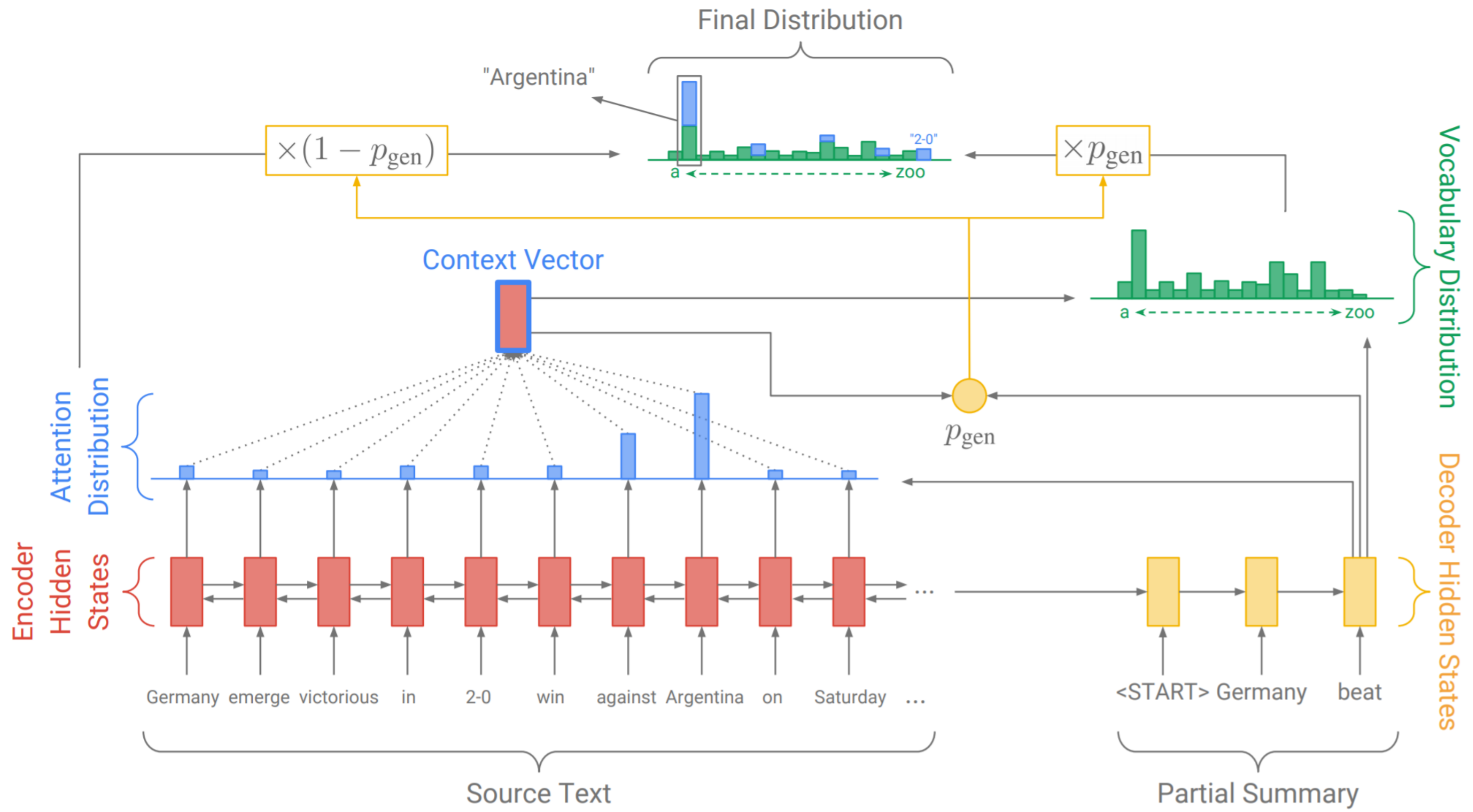
Lionel Messi ganó su primer trofeo interno para Argentina.

We want some parts of the sentence to the copied, some to be translated
Can the network automatically learn to do this?

$$\boxed{P(w) = p_{\text{gen}} P_{\text{vocab}}(w) + (1 - p_{\text{gen}}) \sum_{i:w_i=w} a_i^t}$$

p_{gen}: probability of generating a word from the output vocab

High for a word to be translated, low for a word to be copied

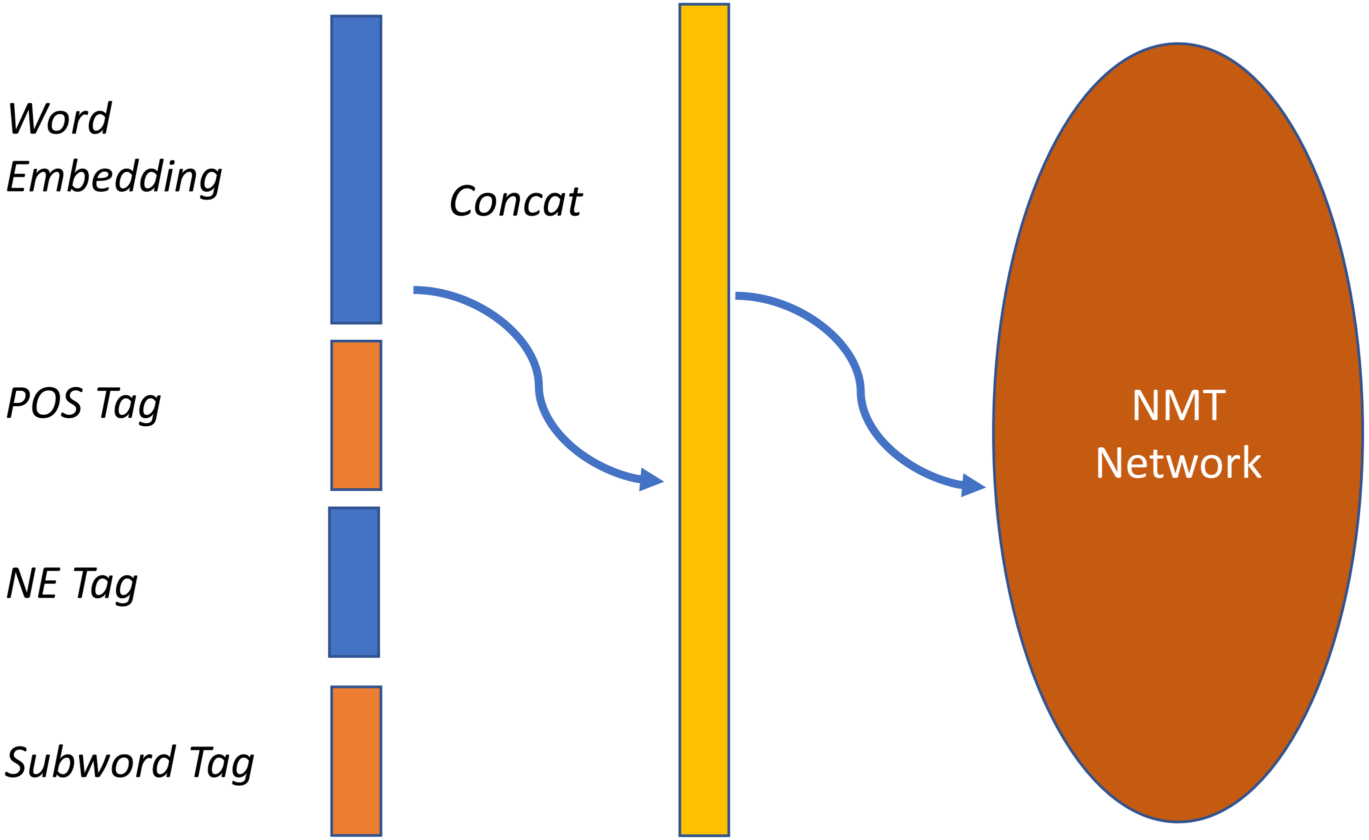


$$p_{gen}^{j} = f(c_j, s_{j-1}, emb(y_{j-1})):$$

Factor-based NMT

Input to the NMT systems is just a sequence of (sub)word embeddings?

Can we provide richer input ➔ more input features?



Additional token Level inputs

Rico Sennrich & Barry Haddow. *Linguistic Input Features Improve Neural Machine Translation*. WMT 2016.

Control tags in the input stream

Special tokens in the input stream to guide to the decoder's generation

Target language *<to_hindi> Argentina won the Copa America tournament.*

Domain *<finance> The bull run is expected to continue for the next three weeks.*

Style *<codemix> <hi> He plays the violin very well.*

These are soft-constraints on the decoder

Training data has to be augmented to understand these special tokens

1. Sennrich, Rico, Barry Haddow, and Alexandra Birch. *Controlling politeness in neural machine translation via side constraints. NAACL* 2016.
2. Johnson, Melvin, et al. "Google's multilingual neural machine translation system: Enabling zero-shot translation." *TACL 2017*.

Multiple input sequences

Some problems need multiple input sequence

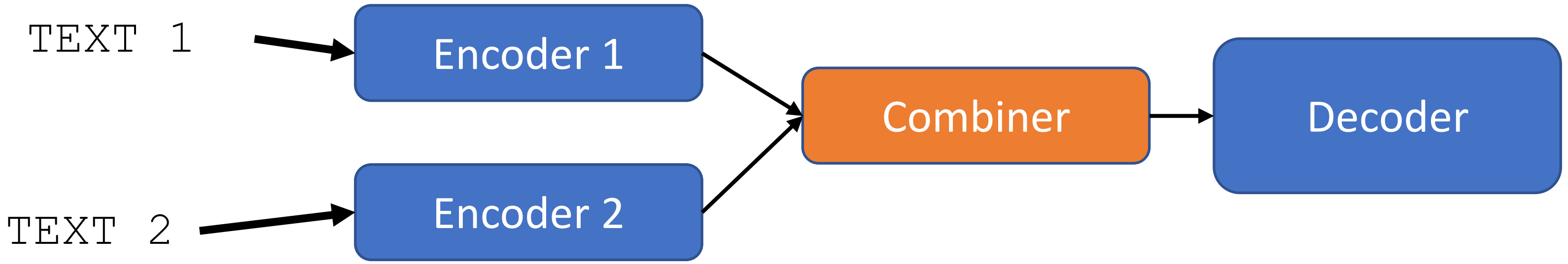
Multi-source translation
(language1, language2)

language3

Automatic Post-editing
(source, MT output)

Post-edited output

Multiple encoders

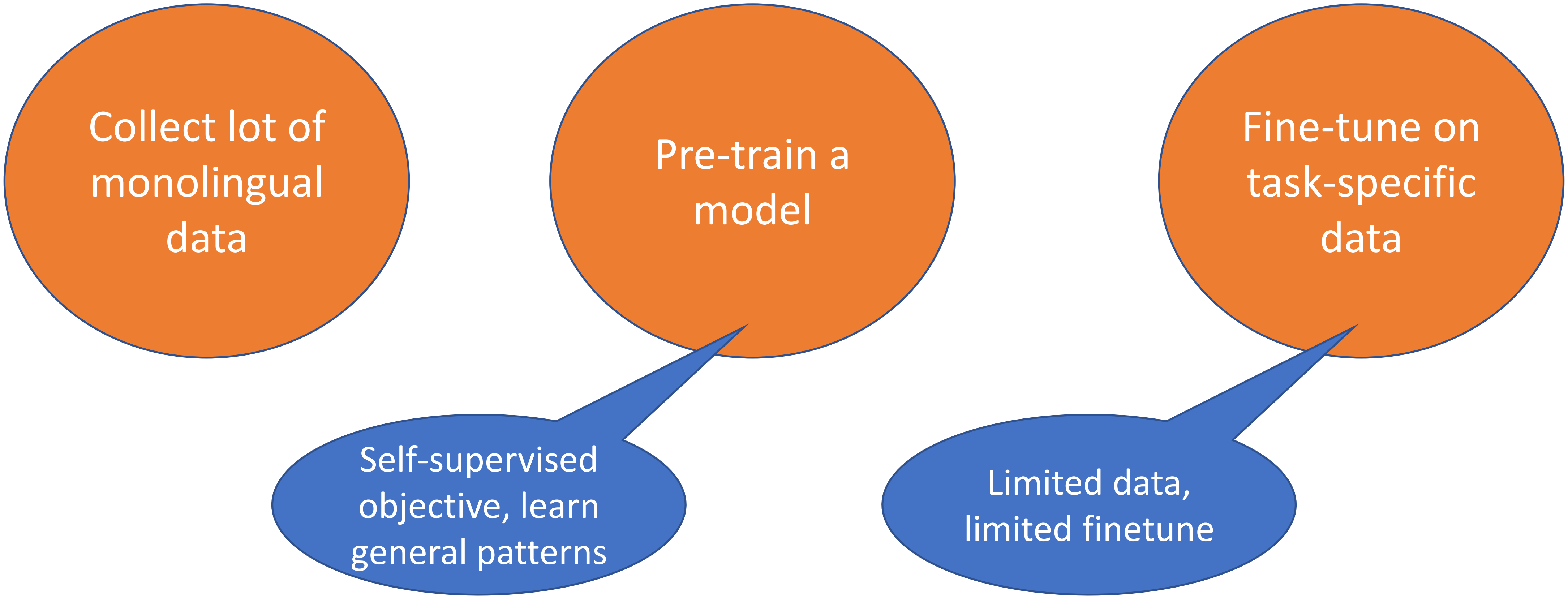


Concat inputs

1. Zoph, Barret and Knight, Kevin. Multi-Source Neural Translation. NAACL 2017.
2. Raj Dabre, Fabien Cromieres, Sadao Kurohashi. Enabling multi-source neural machine translation by concatenating source sentences in multiple languages. Preprint. 2017.

Sequence to Sequence Pre-training

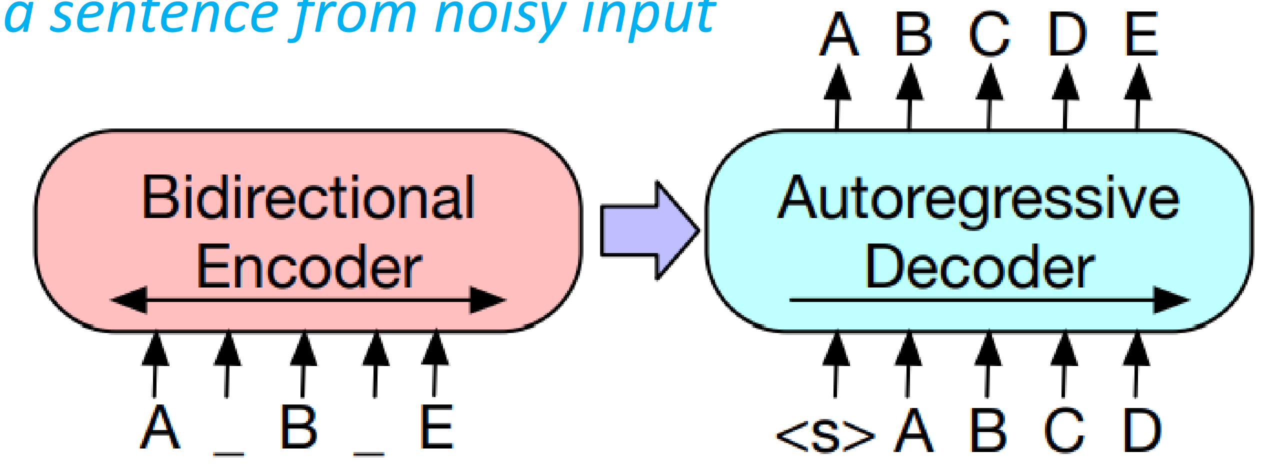
Pre-training has been central to the success of deep-learning based NLP



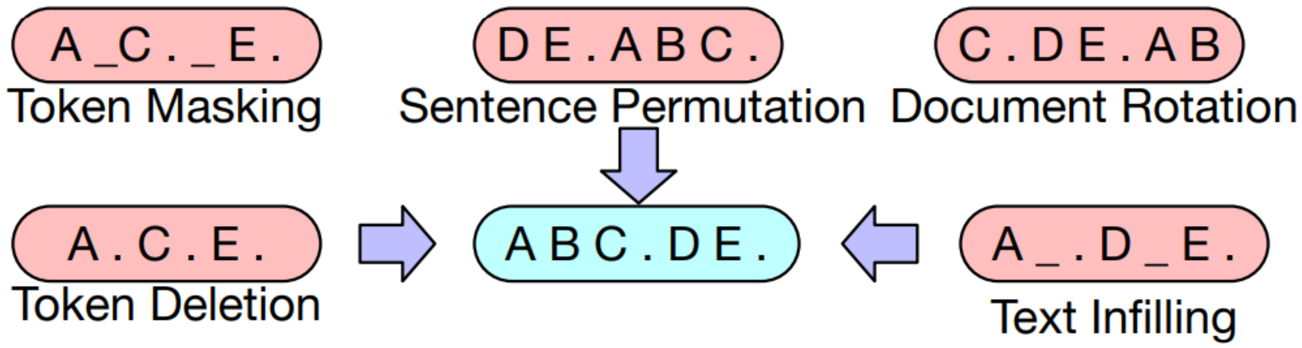
What is the pre-training equivalent to BERT for sequence to sequence models?

Denoising Sequence-to-Sequence Pre-training

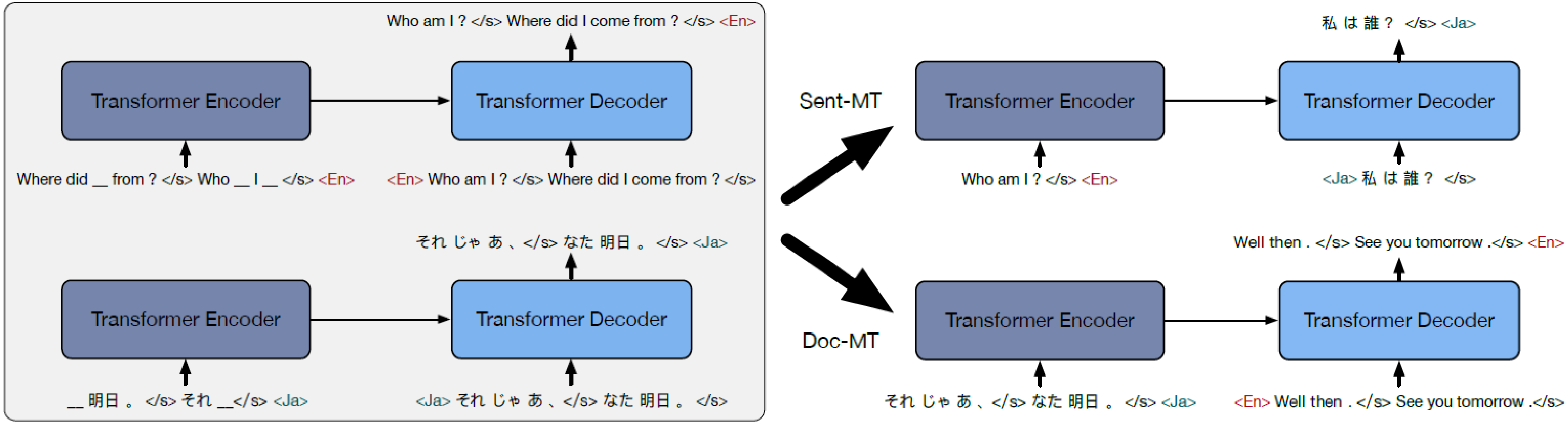
Reconstruct a sentence from noisy input



Different kinds of noise can be added ➔ Span Token masking is most popular



Multilingual pre-training for sequence to sequence models



Multilingual Denoising Pre-Training (mBART) — Fine-tuning on Machine Translation

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, Luke Zettlemoyer.
Multilingual Denoising Pre-training for Neural Machine Translation. TACL. 2020.

Outline

- Introduction
- Statistical Machine Translation
- Neural Machine Translation
- **Evaluation of Machine Translation**
- Multilingual Neural Machine Translation

Evaluation of Machine Translation

Evaluation of MT output

- How do we judge a good translation?
- Can a machine do this?
- Why should a machine do this?
  - Because human evaluation is time-consuming and expensive!
  - Not suitable for rapid iteration of feature improvements

Dimensions of MT Evaluation

- Human evaluation vs. automated metrics
- Quality assessment at sentence (segment) level vs. system level vs. task-based evaluation
- “Black-box” vs. “Glass-box” evaluation

What is a good translation?

Evaluate the quality with respect to:

- **Adequacy**: How good the output is in terms of preserving content of the source text
- **Fluency**: How good the output is as a well-formed target language entity

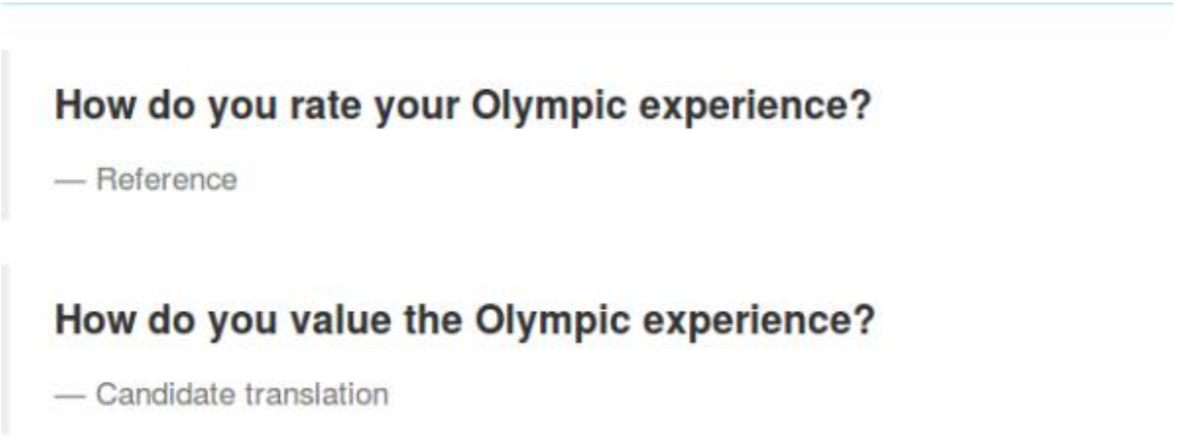
For example, I am attending a lecture

मैं एक व्याख्यान बैठा हूँ
Main ek vyaakhyan baitha hoon
I a lecture sit (Present-first person)
I sit a lecture : Adequate but not fluent

मैं व्याख्यान हूँ
Main vyakhyan hoon
I lecture am
I am lecture: Fluent but not adequate.

Human Evaluation

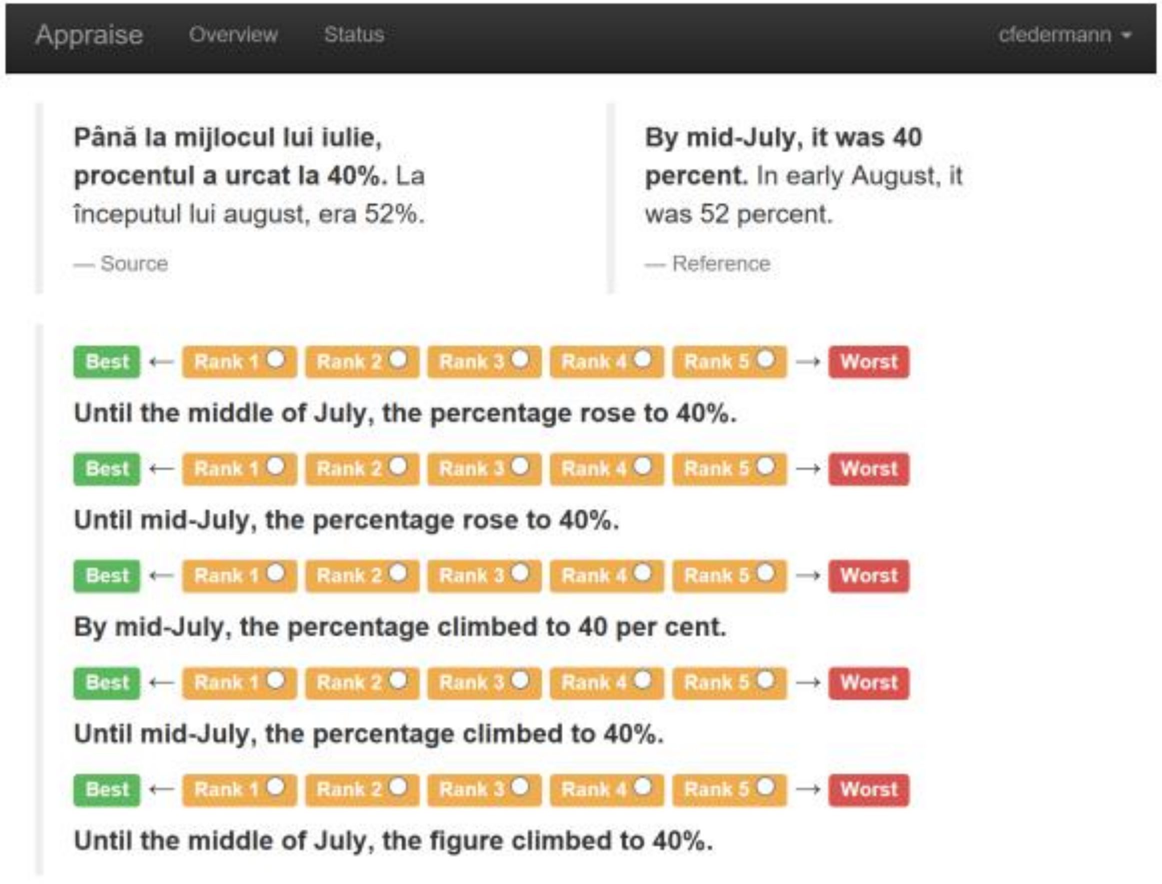
Direct Assessment



Adequacy: Is the meaning translated correctly?
Fluency: Is the sentence grammatically valid?

Adequacy	**Fluency**
5 = All	5 = Flawless
4 = Most	4 = Good
3 = Much	3 = Non-native
2 = Little	2 = Disfluent
1 = None	1 = Incomprehensible

Ranking Translations



$$\text{score}(S_i) = \frac{1}{|\{S\}|} \sum_{S_j \neq S_i} \frac{\text{wins}(S_i, S_j)}{\text{wins}(S_i, S_j) + \text{wins}(S_j, S_i)}$$

Automatic Evaluation

Human evaluation is not feasible in the development cycle

- Given: A corpus of good quality human reference translations
- Output: A numerical “translation closeness” metric
- Given (ref,sys) pair, score = f(ref,sys) ➔ $\mathbb{R}$

  where,

  sys (candidate Translation): Translation returned by an MT system

  ref (reference Translation): ‘Perfect’ translation by humans

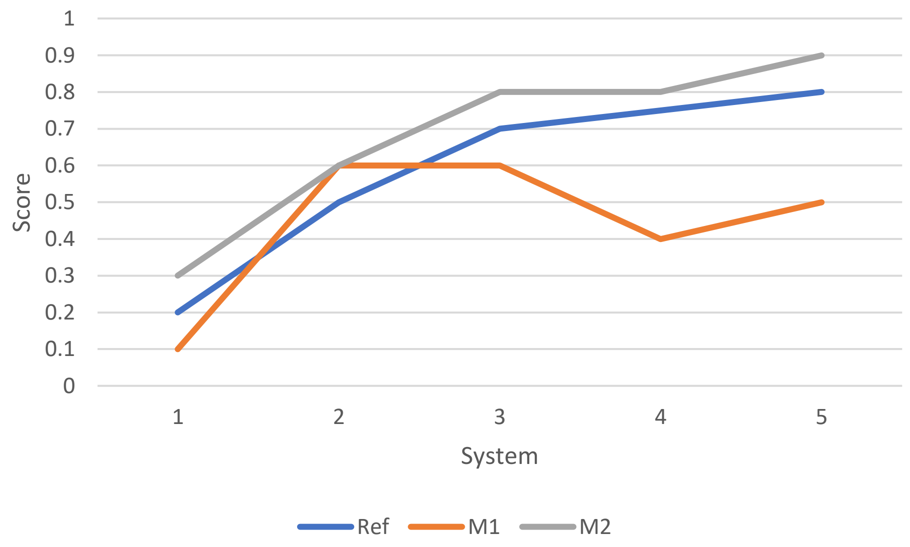
Multiple references are better

Key Idea of Automatic Evaluation

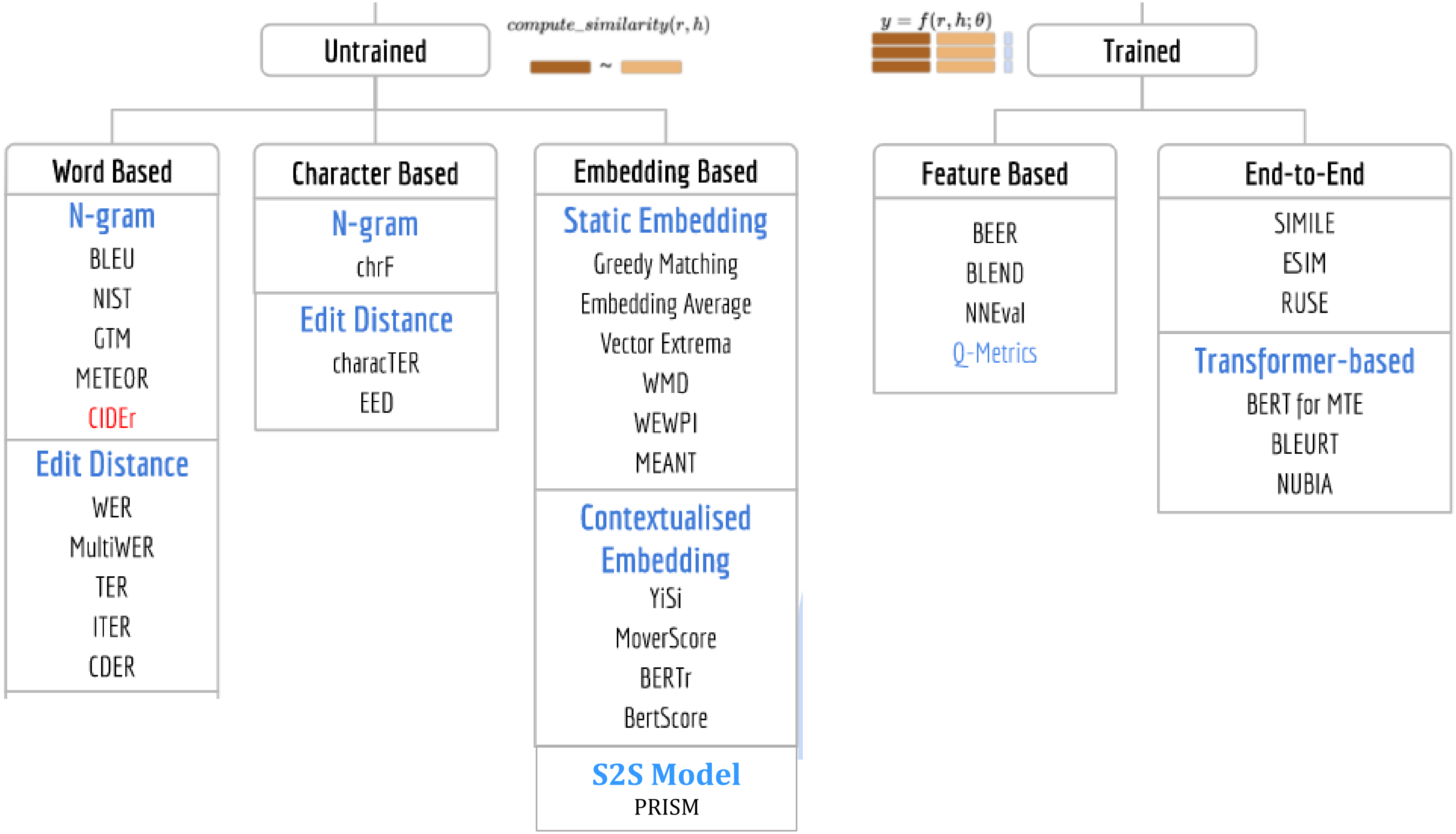
The closer a machine translation is to a professional human translation, the better it is.

How good is an automatic metric?

How well does it correlate with human judgment?



Taxonomy of Evaluation Metrics



Ananya Sai, Akash Kumar Mohankumar, Mitesh Khapra. *A Survey of Evaluation Metrics Used for NLG Systems.* ACM CSUR 2020.

BLEU

(Untrained, word-based, n-gram matching)

- Most popular MT evaluation metric
- Requires only reference translations
  - No additional resources required
- Precision-oriented measure
- Difficult to interpret absolute values
- Useful to compare two systems

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

Candidate 1: **मैने** अब **खाना खाया**
maine ab khana khaya

I *now food ate*
I ate food now

matching unigrams: 3, precision=3/4
matching bigrams: 1, precision=1/3

Weighted average of n-gram precision + Brevity penalty

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a Method for Automatic Evaluation of Machine Translation. ACL 2002.

chrF

(Untrained, character-based, n-gram matching)

- character n-gram F-score
- No additional resources required
- Can address morph-syntactic phenomena
- β controls precision-recall tradeoff ➔ β =2 is widely used

$$\text{CHRF}\beta = (1+\beta^2)\frac{\text{CHRP}\cdot\text{CHRR}}{\beta^2\cdot\text{CHRP}+\text{CHRR}}$$

Maja Popović. chrF: character n-gram F-score for automatic MT evaluation. WMT 2015.

Embedding-based Metrics

Motivation

- Better semantic match
- Address synonym, OOV, etc.

Can use static or contextual embeddings

Sentence Embedding

$$score(p, r) = \cos ine\_sim(\vec{p}, \vec{r})$$

- **Vector Averaging**

$$\vec{s} = \frac{\sum_{w \in s} \vec{w}}{|s|}$$

- **Vector Extrema**

$$\vec{s_d} = \begin{cases} \max_{w \in s} \vec{w}_d, & \text{if } \vec{w}_d > |\min_{w' \in s} \vec{w'}_d| \\ \min_{w \in s} \vec{w}_d & \text{otherwise} \end{cases}$$

- **Direct sentence embeddings** tuned for semantic textual similarity and paraphrasing tasks (e.g. LABSE, sent-transformers)

Soft word-embedding alignment

Greedy Matching

e.g. BERTScore

$$R_{BERT} = \frac{1}{|r|} \sum_{i \in r} \max_{j \in p} \vec{i}^T \vec{j}, \quad P_{BERT} = \frac{1}{|p|} \sum_{j \in p} \max_{i \in r} \vec{i}^T \vec{j}$$

$$\text{BERTscore} = F_{BERT} = 2 \frac{P_{BERT} . R_{BERT}}{P_{BERT} + R_{BERT}}$$

Based on greedily matching a reference word to closest hypothesis word

Optimal Transport

e.g. WMD

$$WMD(p, r) = \min_{T} \sum_{i,j=1}^{n} T_{ij}.\Delta(i, j)$$

$$\text{such that} \sum_{j=1}^{n} T_{ij} = \vec{p}_i \forall i \in \{1, .., n\}, \text{ and } \sum_{i=1}^{n} T_{ij} = \vec{r}_j \forall j \in \{1, .., n\}$$

Based on optimally aligning reference & hypothesis words

PRISM

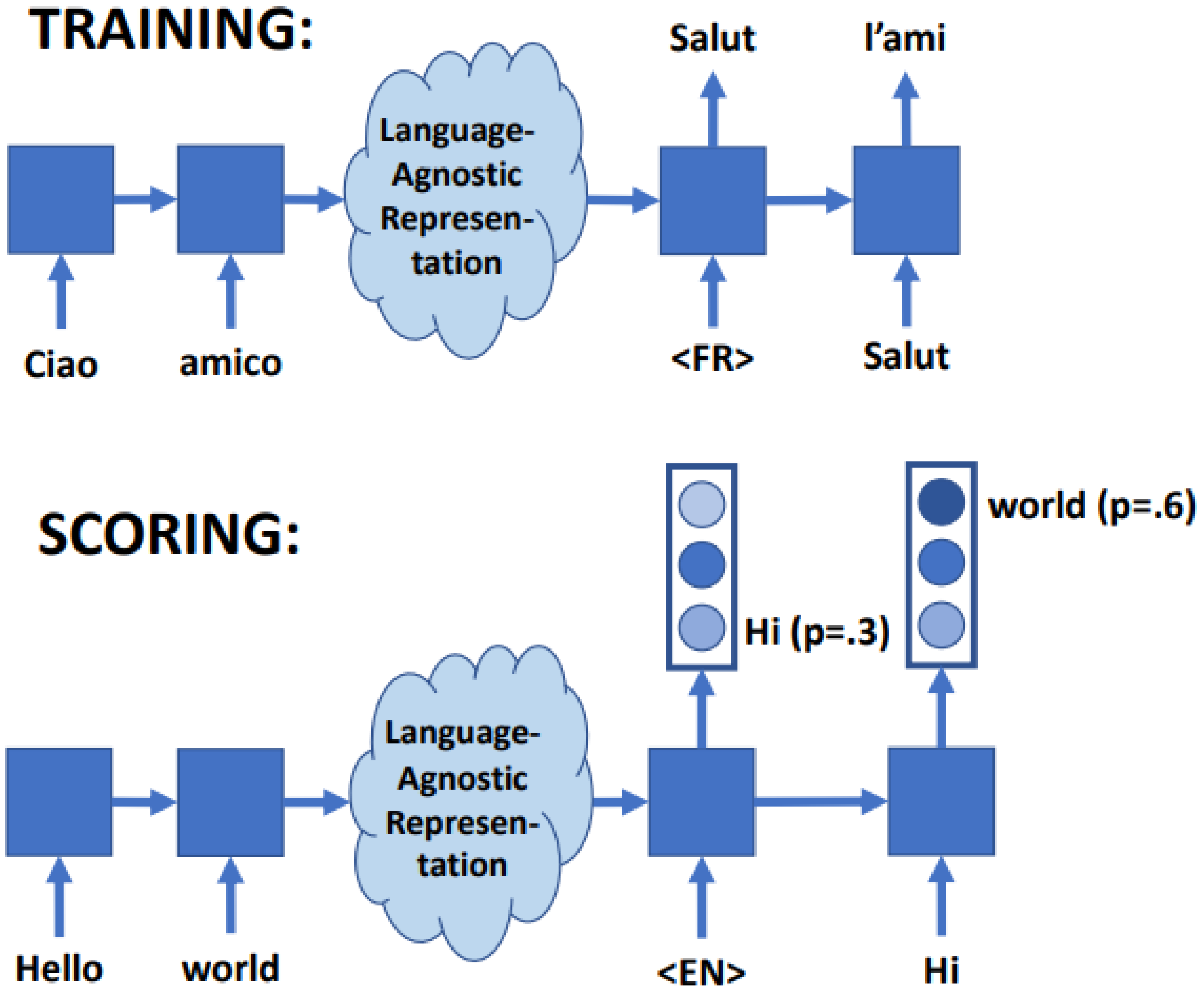
Probability is the metric

(instead of embedding similarity)

Use a multilingual NMT to score (hyp,ref) paiirs

score(hyp,ref) = P(hyp|ref)

NMT system used as a paraphraser



Brian Thompson, Matt Post. *Automatic Machine Translation Evaluation in Many Languages via Zero-Shot Paraphrasing*. EMNLP 2020.

More on PRISM

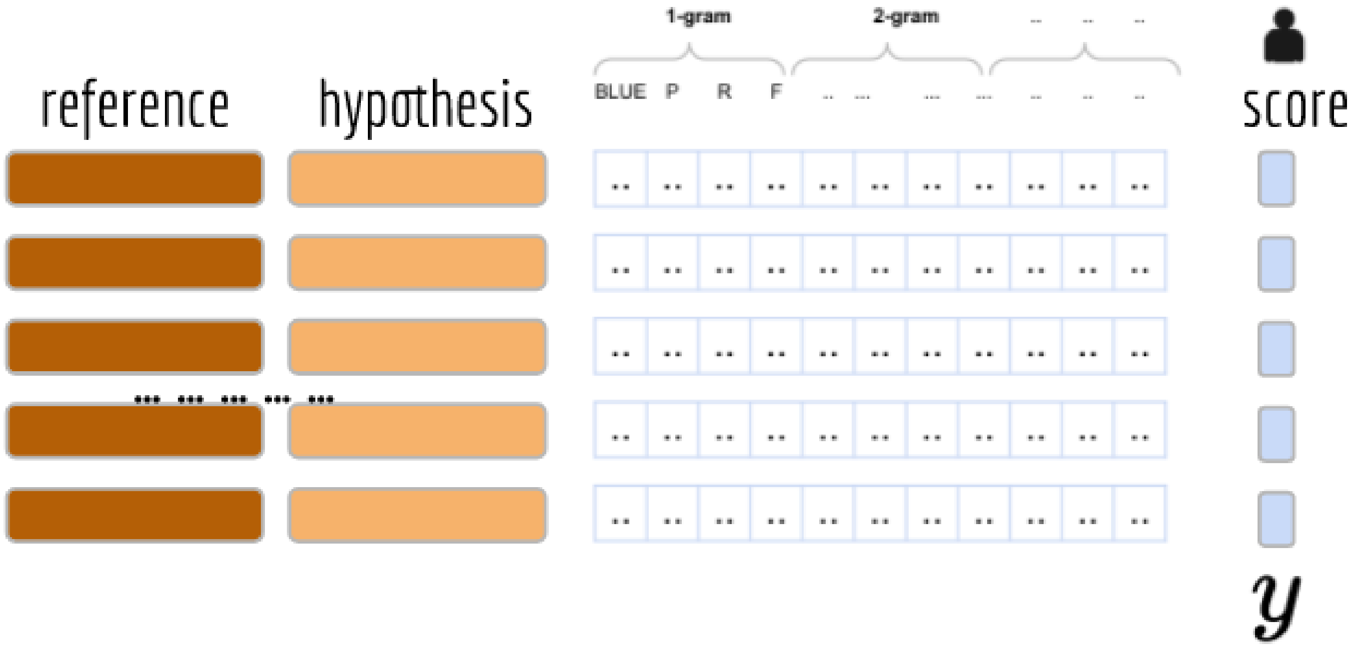
- Length normalize score and average of both directions (hyp ←→ ref)
- Unbiased paraphraser
- When ref is available, doesn’t have to be SOTA MT system
- Ref-based better than ref-free evaluation
- Better than LASER+LM and mBART (mBART is very bad)
- Can distinguish between strong systems also better than other metrics

Trainable Metrics

Learn to assign a score to each hypothesis, reference, (source) tuple

Needs Training data

Database of human judgments
e.g. WMT metrics task data



Feature-based solutions
Lexical, semantic, source/ref, features

End-to-End solutions
Fine-tune pretrain embeddings

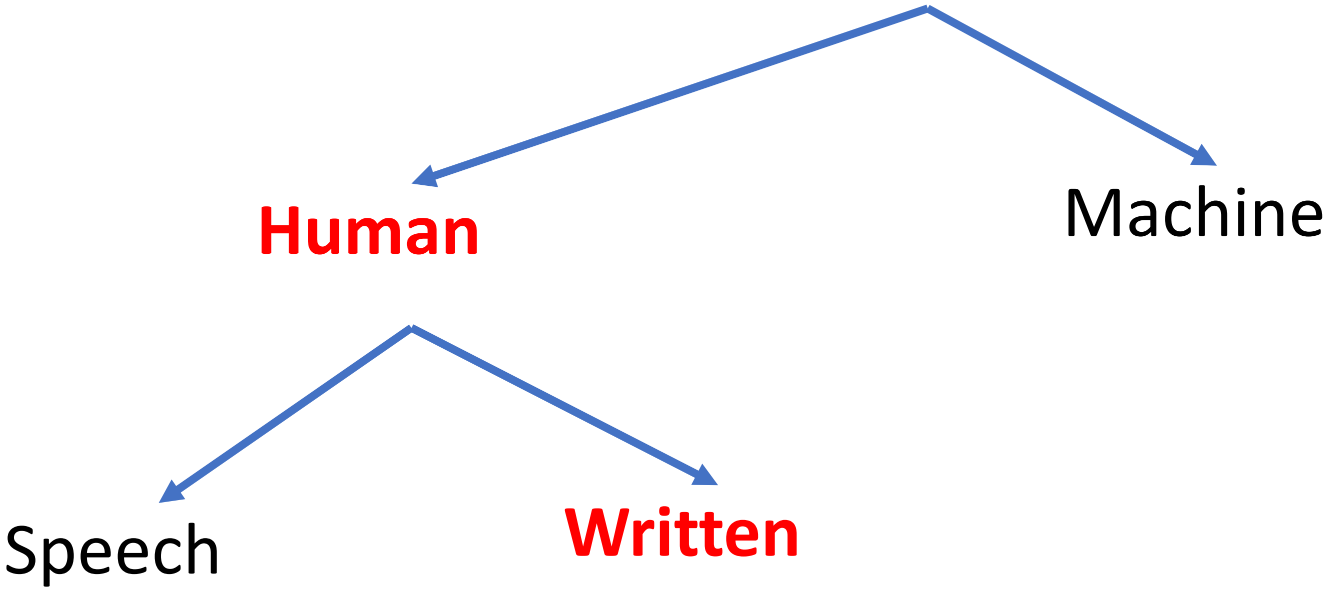
Regression Task $y = f(x; \theta)$

***Metrics**: BEER, BLEND, ESIM, COMET, BLUERT, YiSi-2 etc*

The Effect of Translationese

What is Translationese?

Tanslated text → not originally composed in the language



How is human translationese different?

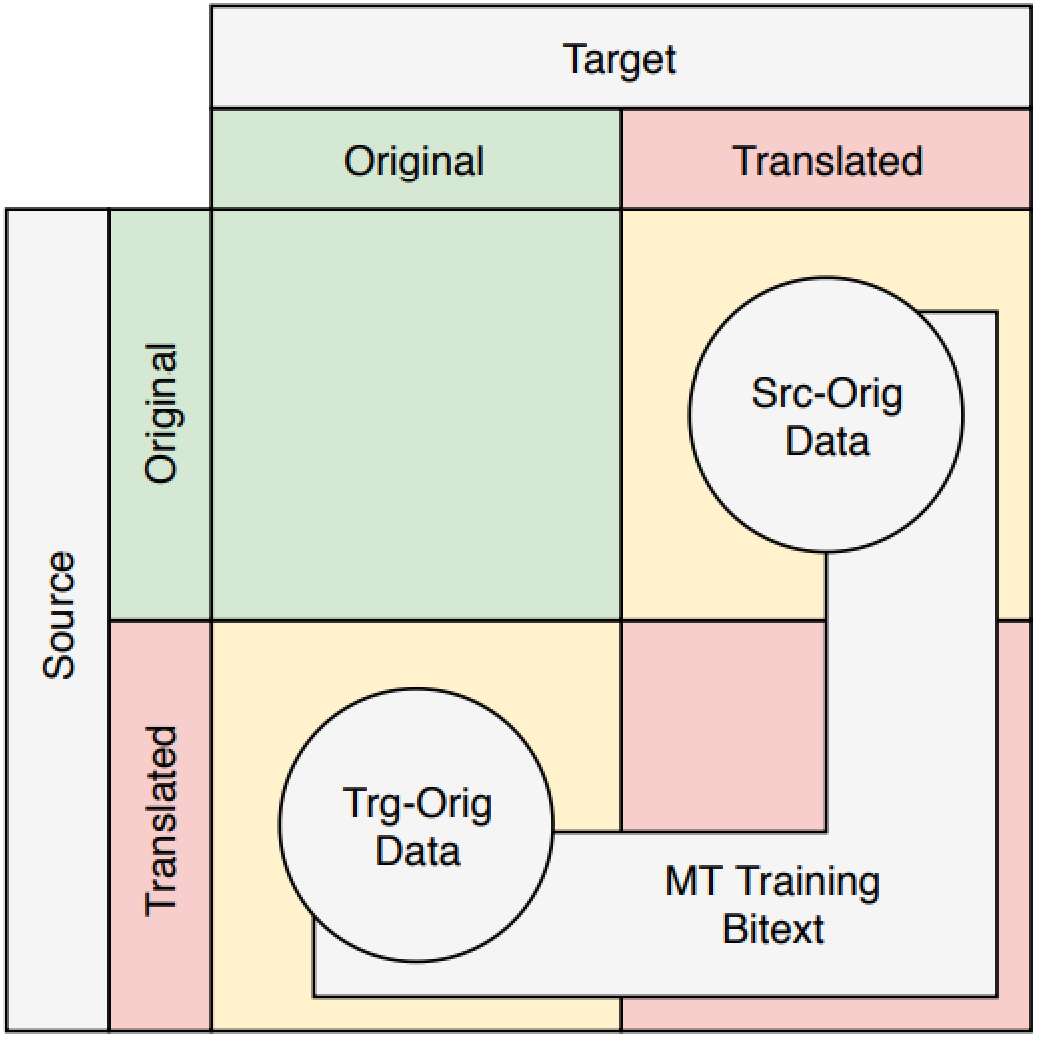
Baker (1993), Toury (2012)

- **Explicitation**: be more explicit than the original source
- **Simplification**: simplified (lexical, syntactically and stylistically);
  - Less ambiguous
- **Normalization**: exaggerate target language features;
  - Unmarked, conventional, less creative, more conservative
  - Conventionalization of metaphors and idioms
  - Dialectical and colloquial expressions less frequent
  - Lexical choice of ‘standard translation’
- **Interference**/Shining through: phenomena pertaining to the make-up of the source text tend to be transferred to the target text

Machine Translationese

(Bizzoni et al., 2020)

- Different from human translationese
- More studies need to characterize difference
- Shining-through obvious
- Literal translations can occur
- Neural MT outputs more complex than SMT output



From Riley et al 2020

Figure 1: Illustration of MT train+test parallel data, organized into quadrants based on whether the source or target is translated or original.

Important Takeaways

(Zhang et al., 2019; Graham et al., 2020 Edunov et al., 2019; Bogoychev et al., 2020)

- Use source original testsets for evaluation
- Relative rankings with TO/mixed testset are largely reliable
- Absolute human scores on TO testset can give an exaggerated indication of translation quality
  - Particularly for low-resource languages
- BLEU scores on TO and SO testsets have issues
  - Human judgment in the only reliable indicator of quality improvement
  - Use LM to evaluate fluency wrt target language model if human evaluation is not feasible

Effect of backtranslation and forward translation

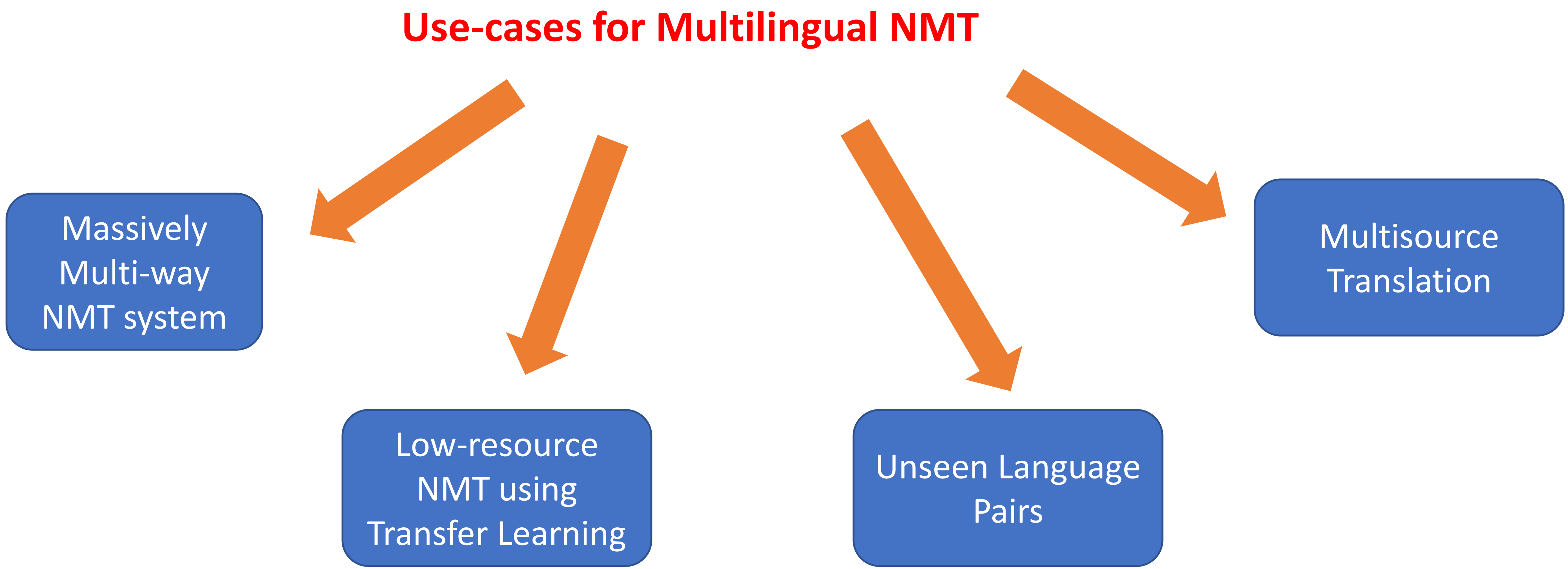
- Backtranslation benefits target original data (BLEU)
- Forward translation benefits source original data (BLEU)
  - Only if original MT system is good
- Reasons:
  - Train/test distribution match
  - BT: target domain adaptation, FT: source domain adaptation
  - In case of BT, simpler input
- Human judgment
  - No significant difference of BT and FT data on adequacy
  - BT data improves fluency

Outline

- Introduction
- Statistical Machine Translation
- Neural Machine Translation
- Evaluation of Machine Translation
- Transformer Architecture
- **Multilingual Neural Machine Translation**

Multilingual Neural Machine Translation

NMT Models involving more than two languages

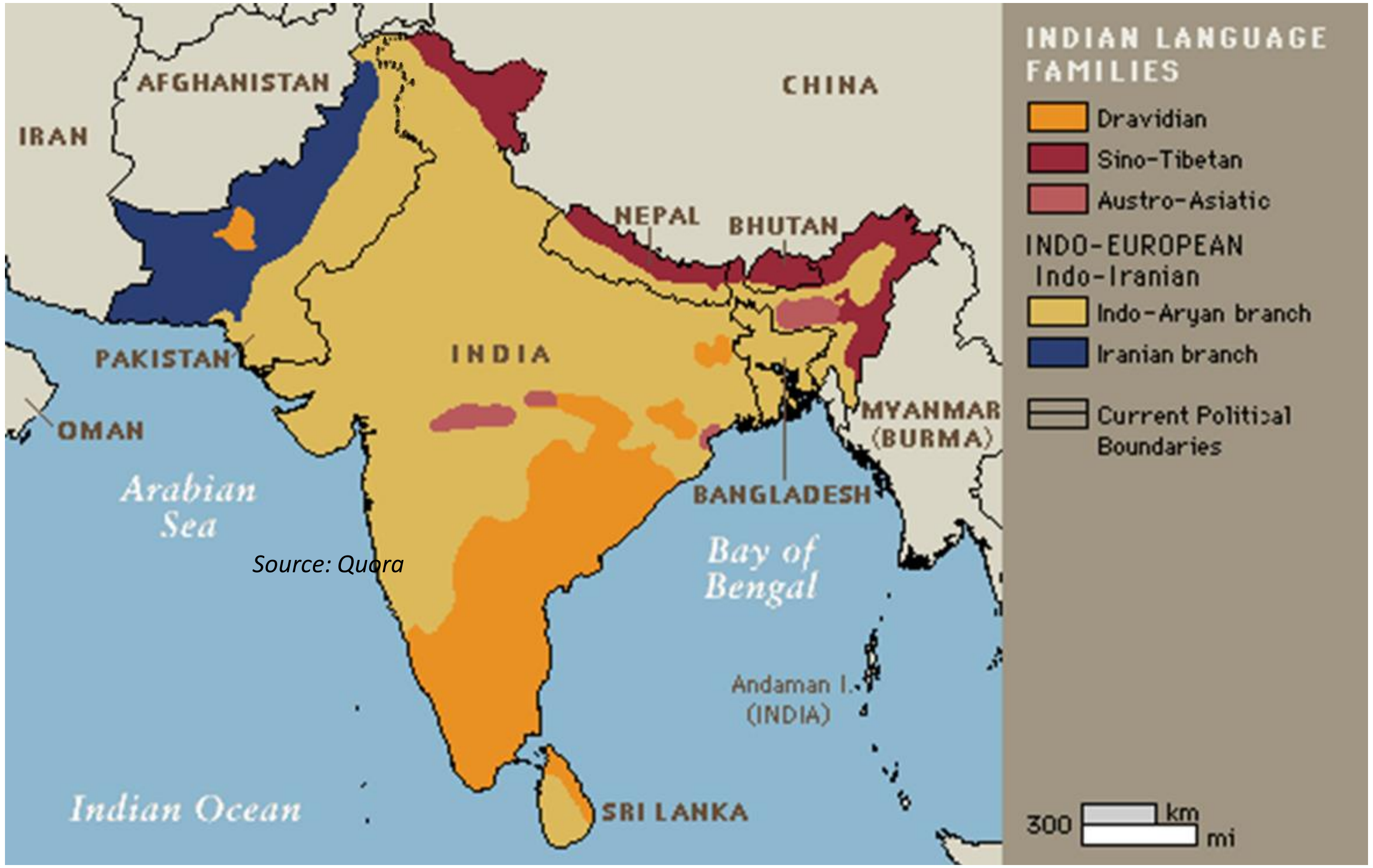


Raj Dabre, Chenhui Chu, Anoop Kunchukuttan. *A Comprehensive Survey of Multilingual Neural Machine Translation.* ACM Computing Surveys. 2020.

Diversity of Indian Languages

Highly multilingual country

Greenberg Diversity Index 0.9



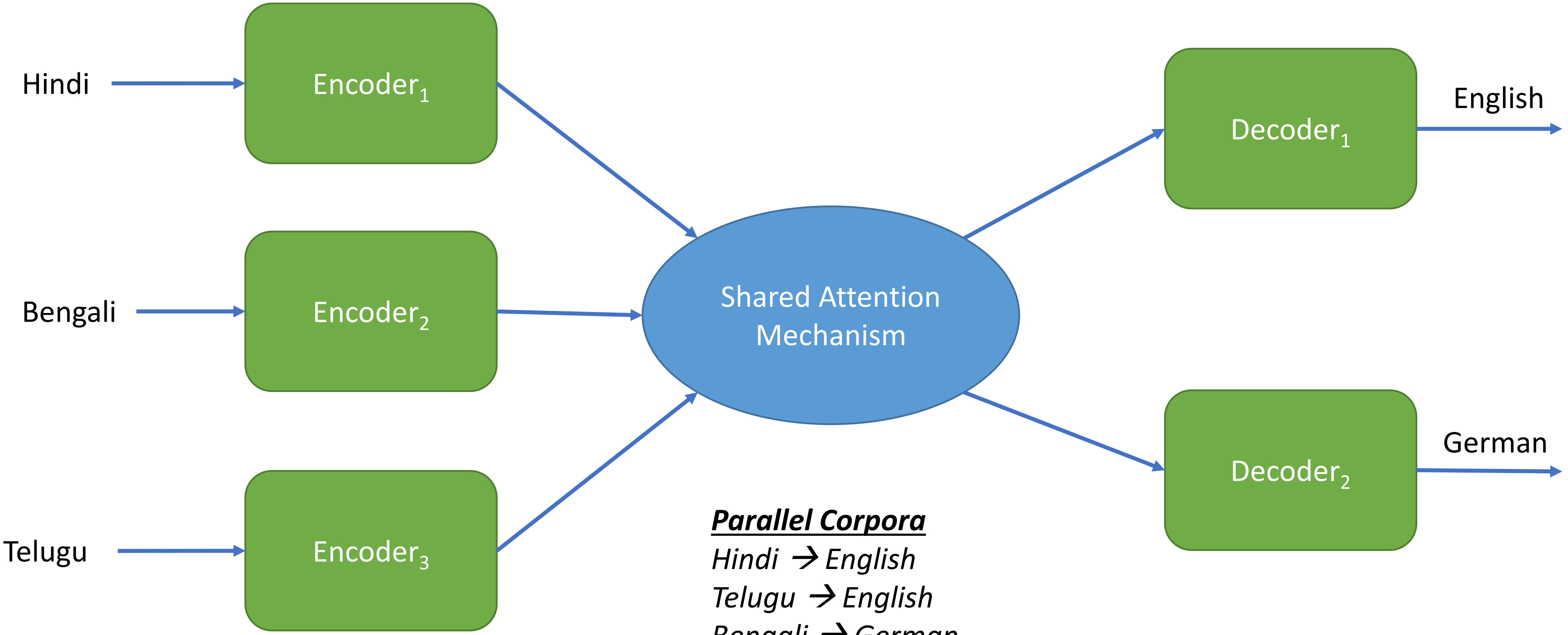
Source: Quora

- *4 major language families*
- *1600 dialects*
- *22 scheduled languages*
- *125 million English speakers*
- *8 languages in the world's top 20 languages*
- *11 languages with more than 25 million speakers*
- *30 languages with more than 1 million speakers*

Sources: Wikipedia, Census of India 2011

General Multilingual Neural Translation

(Firat et al., 2016)



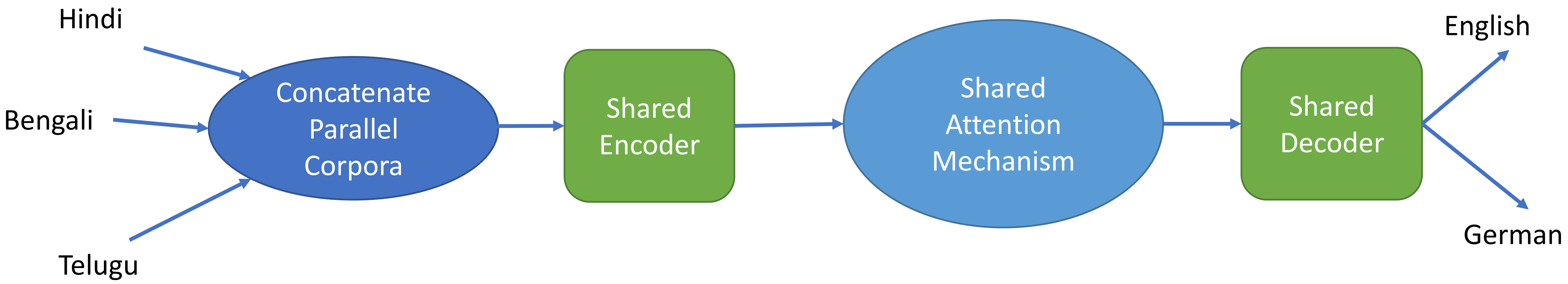
Parallel Corpora

Hindi → English
Telugu → English
Bengali → German

Firat, Orhan, Kyunghyun Cho and Yoshua Bengio. "Multi-way, multilingual neural machine translation with a shared attention mechanism." *NAACL. 2016.*

Compact Multilingual NMT

(Johnson et al., 2017)



Johnson, Melvin, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat et al. "Google's multilingual neural machine translation system: Enabling zero-shot translation." TACL (2017).

Combine Corpora from different languages

(Nguyen and Chang, 2017)

I am going home	હુ ઘરે જવ છૂ
It rained last week	છેલ્લા આઠવડિયા મા વર્સાદ પાડ્યો

It is cold in Pune	पुण्यात थंड आहे
My home is near the market	माझा घर बाजाराजवळ आहे

Convert Script

Concat Corpora

I am going home	हु घरे जव छू
It rained last week	छेल्ला आठवडिया मा वर्साद पाड्यो
It is cold in Pune	पुण्यात थंड आहे
My home is near the market	माझा घर बाजाराजवळ आहे

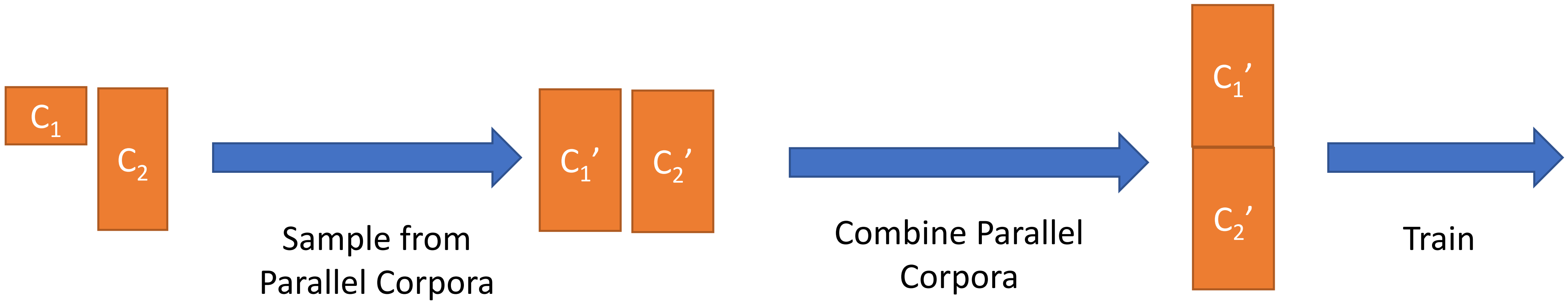
There is only one decoder, how do we generate multiple languages?

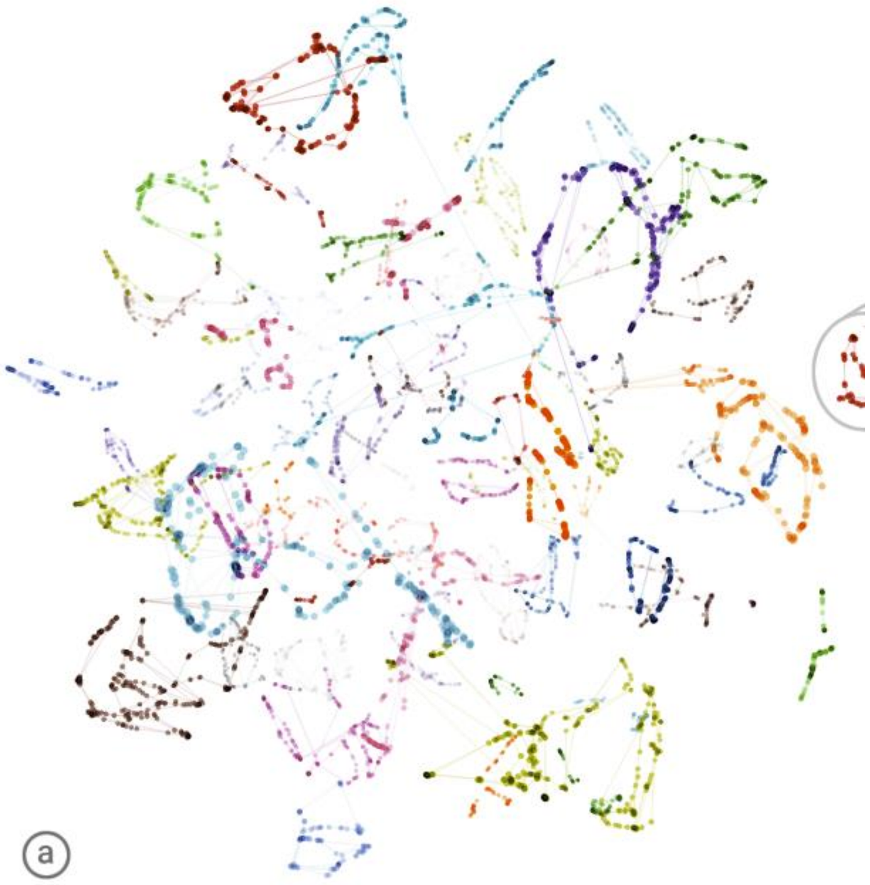
Language Tag Trick → Special token in input to indicate target language

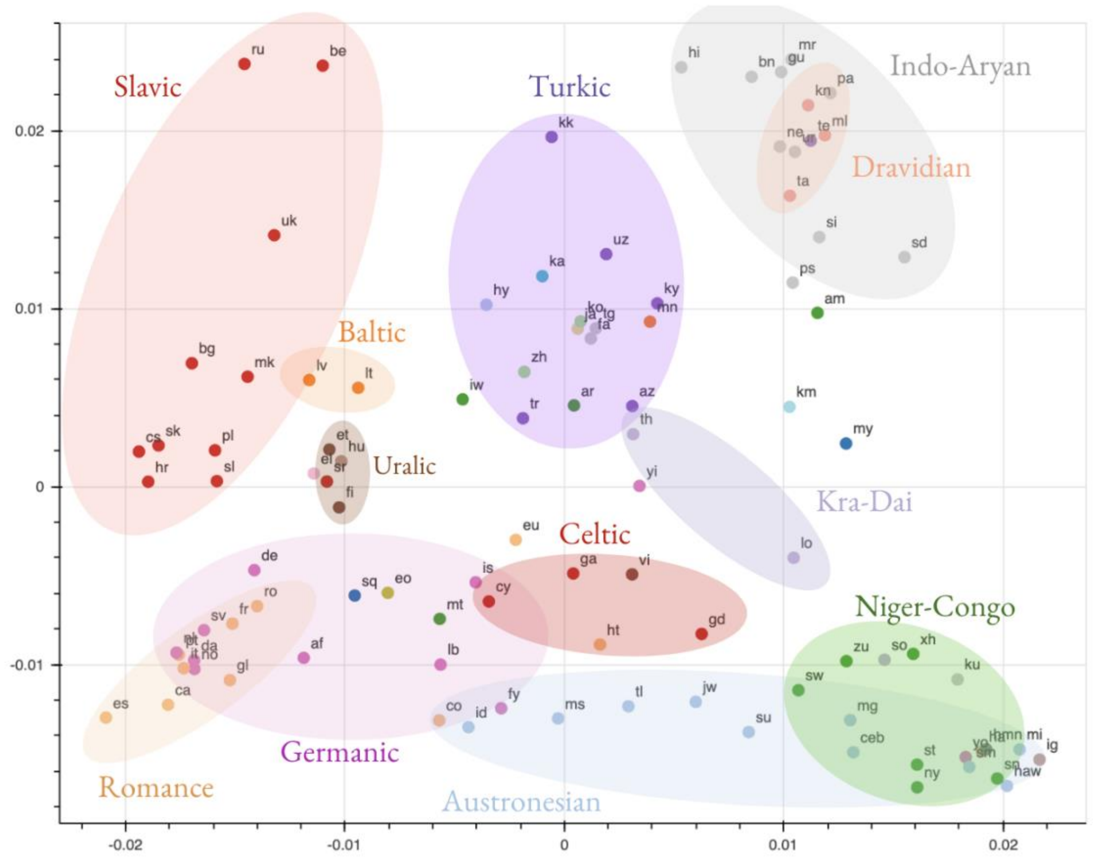
Original Input: *मकर संक्रांति भगवान सूर्य के मकर में आने का पर्व है*

Modified Input: *मकर संक्रांति भगवान सूर्य के मकर में आने का पर्व है<eng>*

Joint Training







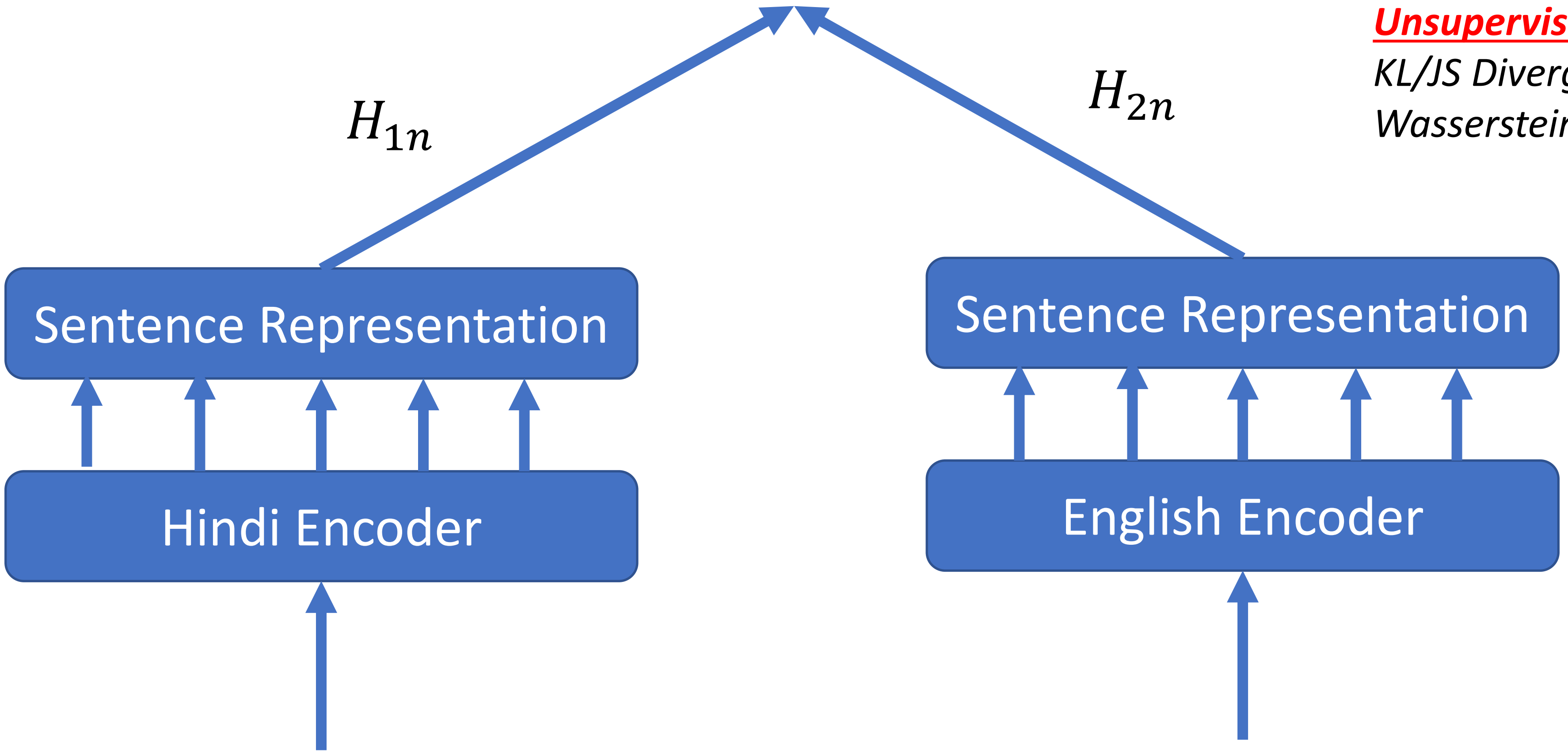
Similar sentences have similar encoder representations

But the multilingual representation is not perfect

Learning common representations across languages is one of the central problems for multilingual NMT

Aligning Encoder Representations

$$\min_{\theta} \sum_{n=1}^{N} dist(H_{1n}(\theta), H_{2n}(\theta))$$



Supervised Distance Functions

Cosine

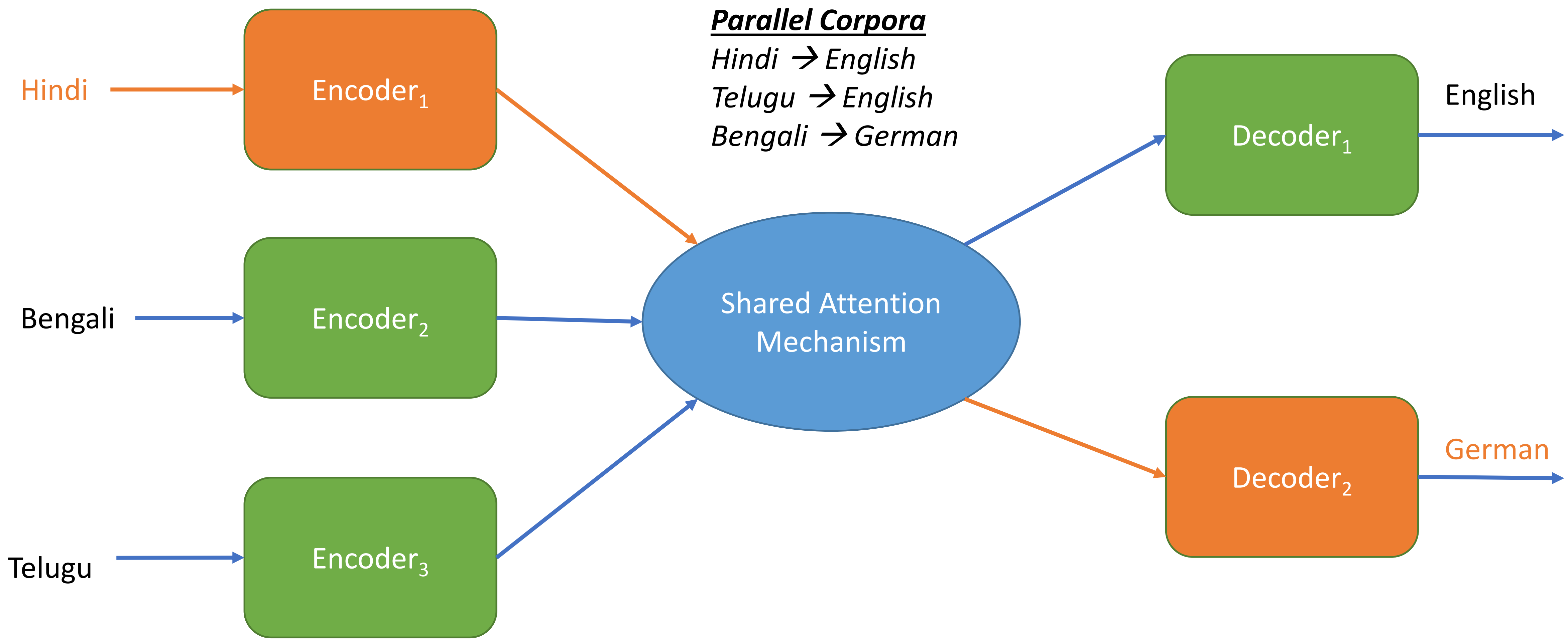
Correlation

Euclidean Distance

Unsupervised Distance Functions

KL/JS Divergence

Wasserstein



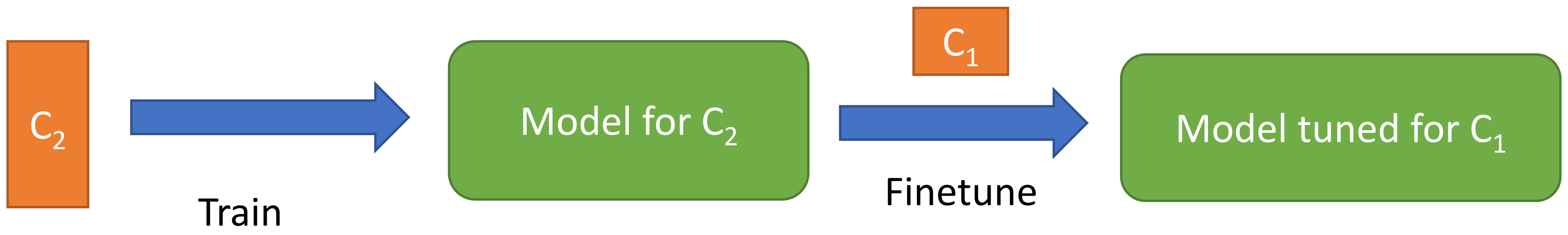
Multilingual NMT makes possible translation between unseen pairs

Zeroshot NMT (Johnson et al., 2017)

Transfer Learning

We want Gujarati → English translation ➔ but little parallel corpus is available

We have lot of Marathi → English parallel corpus



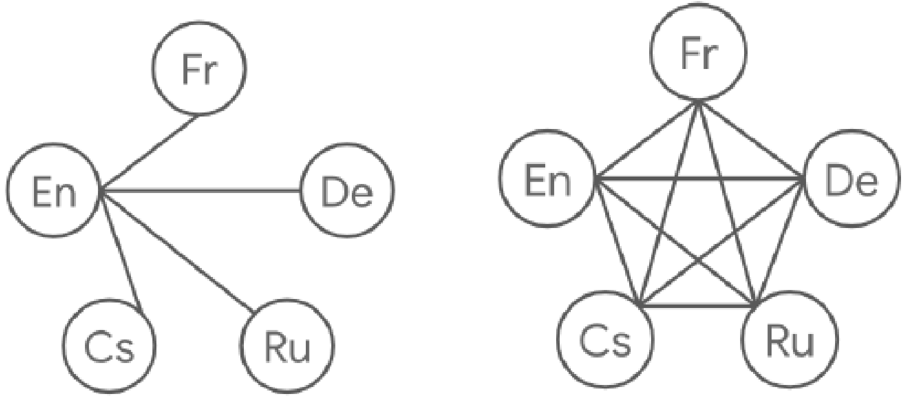
Transfer learning works best for related languages

Complete NMT Translation

WMT data (6 languages, 30 pairs)

	cs	de	en	es	fr	ru
cs		0.7	47	0.8	1	0.9
de	0.7		4.5	2.3	2.5	0.3
en	47	4.5		13.1	38.1	33.5
es	0.8	2.3	13.1		10	4.4
fr	1	2.5	38.1	10		4.8
ru	0.9	0.3	33.5	4.4	4.8	

Mine non-English centric corpora from English-centric parallel corpora



(a) English-centric (b) Complete

Samanantar Corpus: ~80m Indic-Indic sentences

X-Y | *Bleib sicher* ↔ Stay safe
Z-Y | Mantente segura ↔ Stay safe
X-Y-Z | *Bleib sicher* ↔ Mantente segura ↔ Stay safe

Complete NMT models can outperform zeroshot and pivot translation models

Markus Freitag and Orhan Firat. Complete Multilingual Neural Machine Translation. WMT 2020.

Summary

Summary

- Machine Translation is one of the most challenging and exciting NLP problems
  - Watch out for advances in MT!
- Machine Translation is important to build multilingual NLP systems
- NMT has been a great success story for Deep Learning
- NMT has the following benefits
  - Improved Fluency & better Word Order
  - Opens up new avenues: Transfer learning, Unsupervised NMT, Zeroshot NMT

Important Takeaways

- Why is MT challenging?
- Word Alignment
- Sequence to Sequence Tasks
- Encoder-Decoder architectures
- Attention Networks
- Transformer Architecture
- Decoding with Beam Search
- Subword vocabulary
- Multilingual/Multitask S2S Models
- MT Evaluation is a challenge

More Reading Material

This was a small introduction, you can find mode elaborate presentations, books and further references below:

SMT Tutorials & Books

- *Machine Learning for Machine Translation (An Introduction to Statistical Machine Translation)*. **Tutorial at ICON 2013** [slides]
- *Machine Translation: Basics and Phrase-based SMT*. **Talk at the Ninth IIIT-H Advanced Summer School on NLP (IASNLP 2018), IIIT Hyderabad** . [pdf]
- Statistical Machine Translation. Philip Koehn. Cambridge University Press. 2008. [site]
- Machine Translation. Pushpak Bhattacharyya. CRC Press. 2015. [site]

NMT Tutorials & Books

- Neural Machine Translation and Sequence-to-sequence Models: A Tutorial. Graham Neubig. 2017. [pdf]
- CMU CS 11-731, Fall 2019 - Machine Translation and Sequence-to-Sequence Models. [link]
- Neural Machine Translation: A Review and Survey. Felix Stahlberg. JAIR. 2020. [link]
- Raj Dabre, Chenhui Chu, Anoop Kunchukuttan. A Comprehensive Survey of Multilingual Neural Machine Translation. ACM Computing Surveys. 2020. [link]

Other Lectures

- https://github.com/oxford-cs-deepnlp-2017/lectures (Lectures 7 & 8)
- https://www.cse.iitm.ac.in/~miteshk/CS6910.html (Lectures 16)
- http://web.stanford.edu/class/cs224n/ (Lectures 7)

Tools

- **moses**: A production-quality open source package for SMT
- **fairseq**: Modular and high-performance NMT system based on PyTorch
- **openNMT-pytorch**: Modular NMT system based on PyTorch
- **marian**: High-performance NMT system written in C++
- **subword-nmt**: BPE tokenizer
- **sentencepiece**: Subword tokenizer implementing BPE and word-piece
- **indic-nlp-library**: Python library for processing Indian language datasets
- **sacrebleu**: MT evaluation tool

Datasets

- Workshop on Machine Translation datasets
- Workshop on Asian Translation datasets
- Samanantar Parallel Corpus
- IITB English-Hindi Parallel Corpus
- ILCI parallel corpus
- WAT2021-Indic Languages Multilingual Parallel
- FLORE-101 testset

More parallel corpora and resources for Indian languages can be found here:

https://github.com/indicnlpweb/indicnlp_catalog

Thank You!

anoop.kunchukuttan@gmail.com

http://anoopk.in

Extra Study Material

Phrase-based SMT Enhancements

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

- **Divergent Word Order**
- Rich morphology
- Named Entities and Out-of-Vocabulary words

Getting word order right

Phrase based MT is not good at learning word ordering

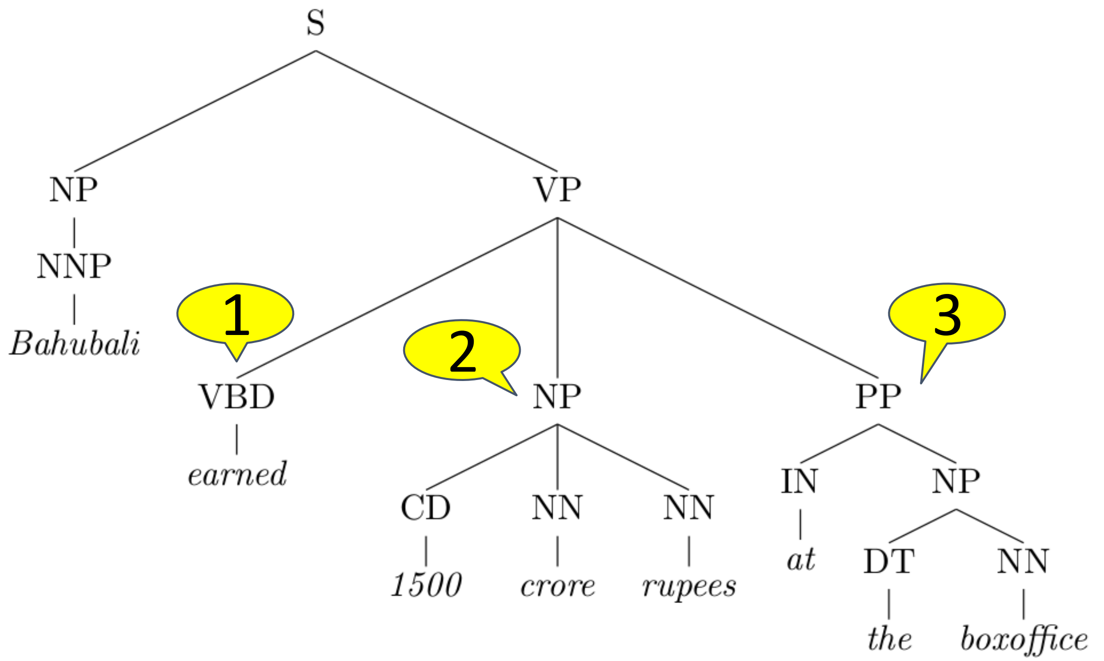
Solution: Let's help PB-SMT with some preprocessing of the input

Change order of words in input sentence to match order of the words in the target language

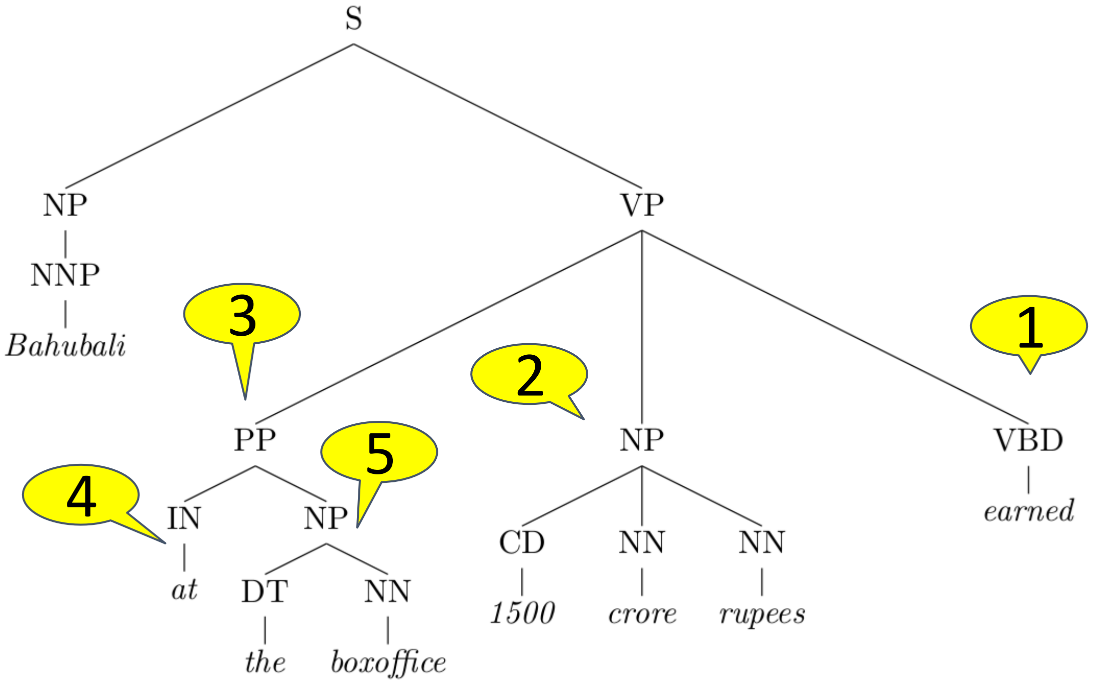
Let's take an example

Bahubali earned more than 1500 crore rupees at the boxoffice

Parse the sentence to understand its syntactic structure



Apply rules to transform the tree

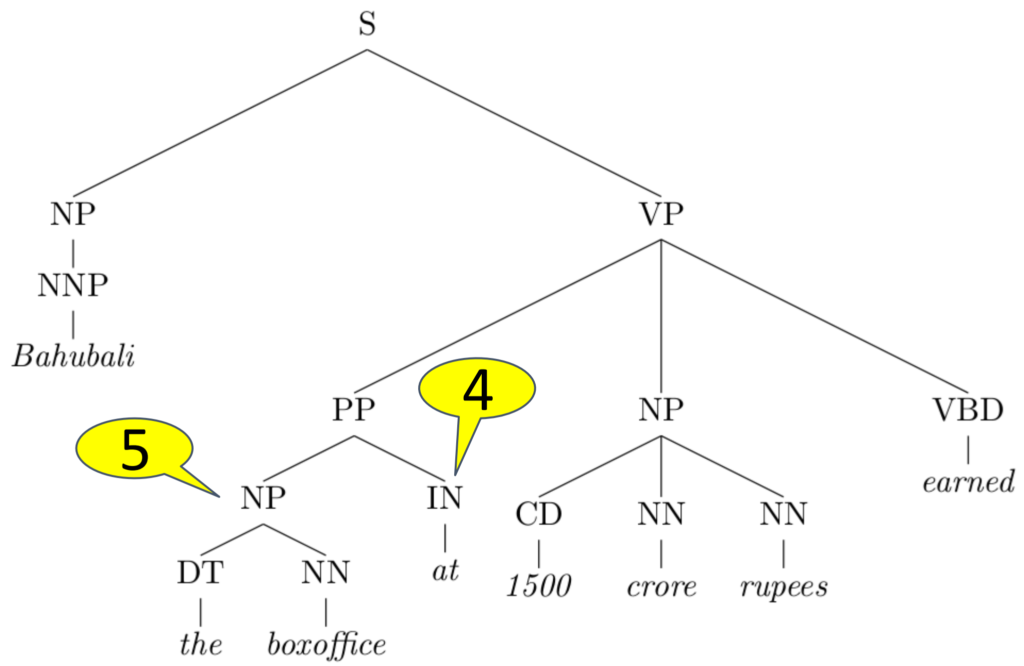


VP → VBD NP PP ⇒ VP → PP NP VBD

This rule captures Subject-Verb-Object to Subject-Object-Verb divergence

Prepositions in English become postpositions in Hindi

PP → IN NP ⇒ PP → NP 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 बॉक्सओफिस पर 1500 करोड रुपए कमाए

These rules can be written manually or learnt from parse trees

Addressing Rich Morphology

Inflectional forms of the Marathi word घर

घर	house
घरात	in the house
घरावरती	on the house
घराखाली	below the house
घरामध्ये	in the house
घरामागे	behind the house
घराचा	of the house
घरामागचा	that which is behind the house
घरासमोर	in front of the house
घरासमोरचा	that which is in front of the house
घरांसमोर	in front of the houses

Hindi words with the suffix वाद

साम्यवाद	communism
समाजवाद	socialism
पूंजीवाद	capitalism
जातीवाद	casteism
साम्राज्यवाद	imperialism

The corpus should contains all variants to learn translations

This is infeasible!

Language is very productive, you can combine words to generate new words

Addressing Rich Morphology

Inflectional forms of the Marathi word घर

घर	house
घर ◌ा त	in the house
घर ◌ा वरती	on the house
घर ◌ा खाली	below the house
घर ◌ा मध्ये	in the house
घर ◌ा मागे	behind the house
घर ◌ा चा	of the house
घर ◌ा माग चा	that which is behind the house
घर ◌ा समोर	in front of the house
घर ◌ा समोर चा	that which is in front of the house
घर ◌ा ◌ं समोर	in front of the houses

Hindi words with the suffix वाद

साम्य वाद	communism
समाज वाद	socialism
पूंजी वाद	capitalism
जाती वाद	casteism
साम्राज्य वाद	imperialism

- *Break the words into its component morphemes*
- *Learn translations for the morphemes*
- *Far more likely to find morphemes in the corpus*

Handling Names and OOVs

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?
What we want to do is map the Roman characters to Devanagari to they sound the same when read → सचिन तेंदुलकर
*➔ We call this process **'transliteration'***

Can be seen as a simple translation problem at character level with no re-ordering

s a c h i n →स च ि न