Machine Translation for Related Languages

Anoop Kunchukuttan

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

Microsoft AI and Research, Hyderabad

Microsoft

BITS Pilani, Hyderabad Campus, February 2019

Outline

- Introduction to Statistical Machine Translation
- Introduction to Neural Machine Translation
- Machine Translation for Related Languages
- Multilingual Learning

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>

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 ➔ *the great diversity among languages of the world*

The central problem of MT is to bridge this language divergence

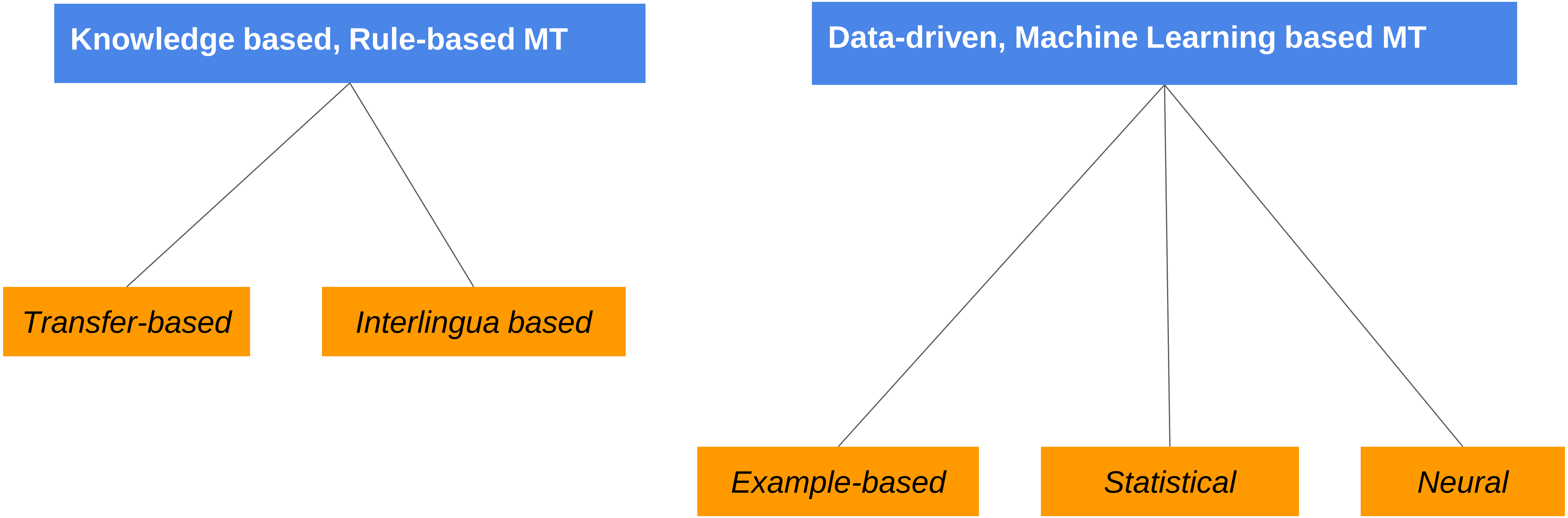
Why is Machine Translation difficult?

- **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

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



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: source language sentence
f : target 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

Let's assume we know how to learn n-gram language models

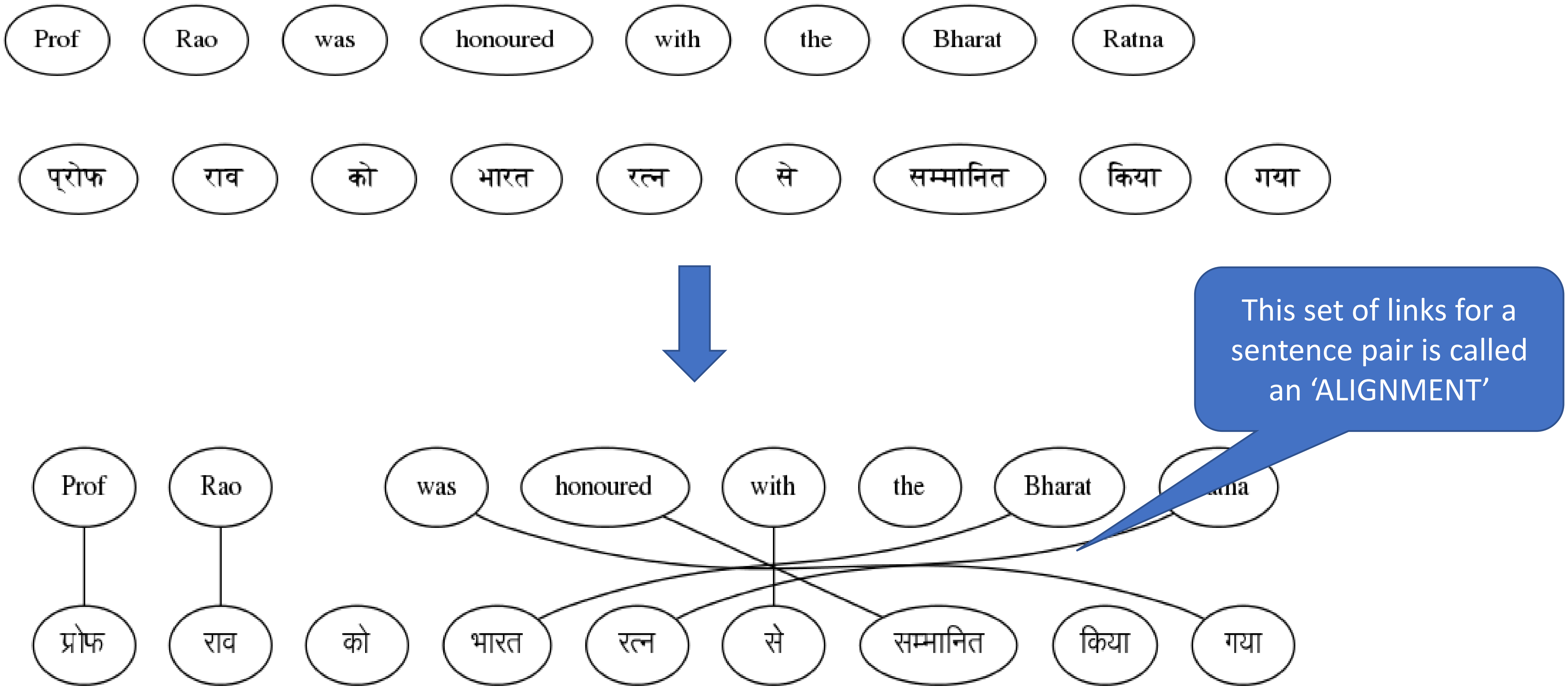
Let's see how to learn the translation model → $P(\boldsymbol{f}|\boldsymbol{e})$

To learn sentence translation probabilities,

→ we first need to learn word-level translation probabilities

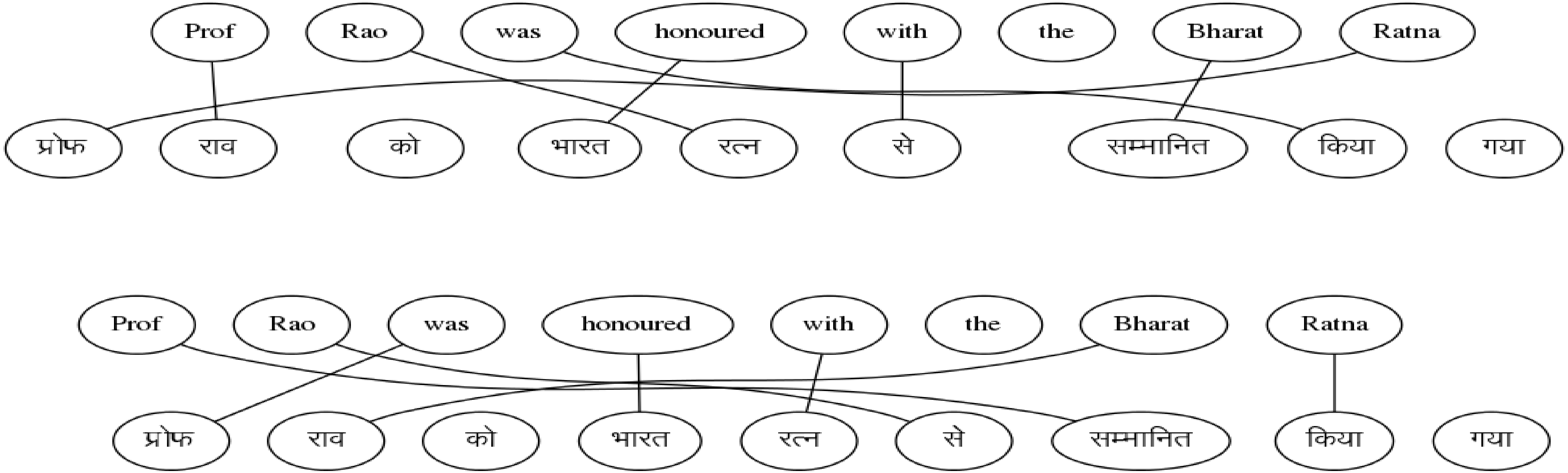
That is the task of word alignment

Given a parallel sentence pair, find word level correspondences



But there are multiple possible alignments

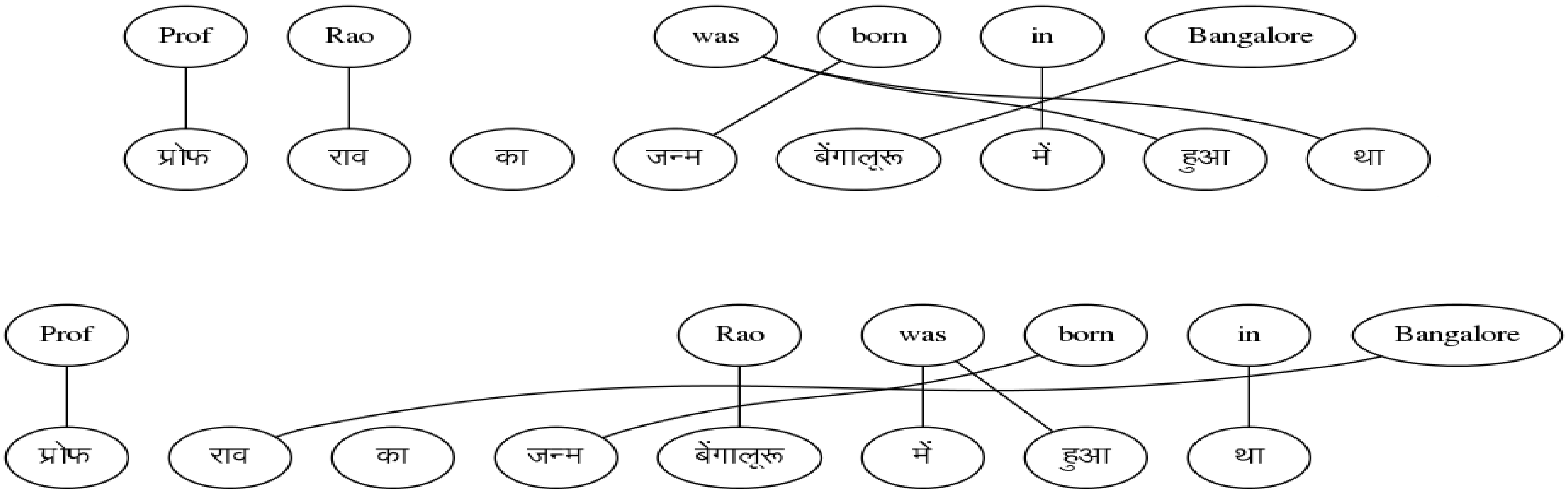
Sentence 1



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 बैठी है

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

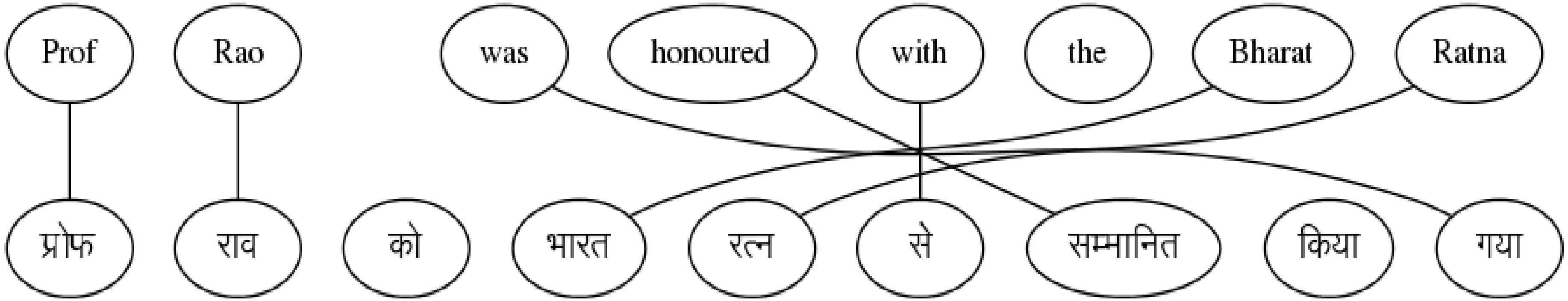
Key Idea

Co-occurrence of translated words

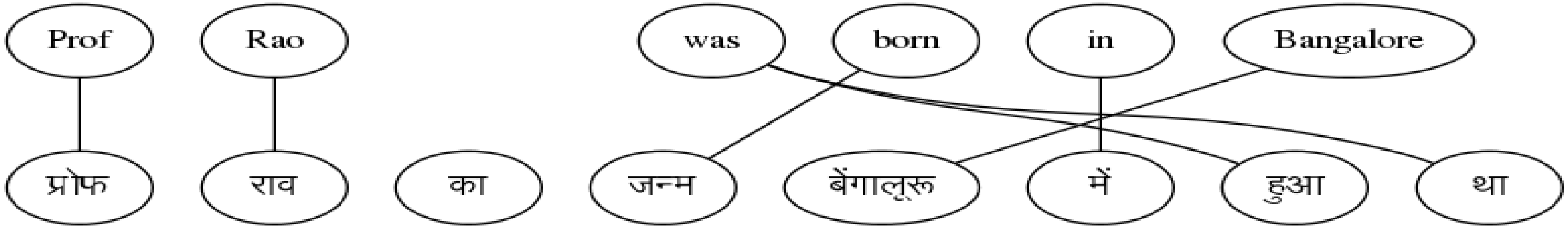
Words which occur together in the parallel sentence are likely to be translations (higher P(f|e))

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

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

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

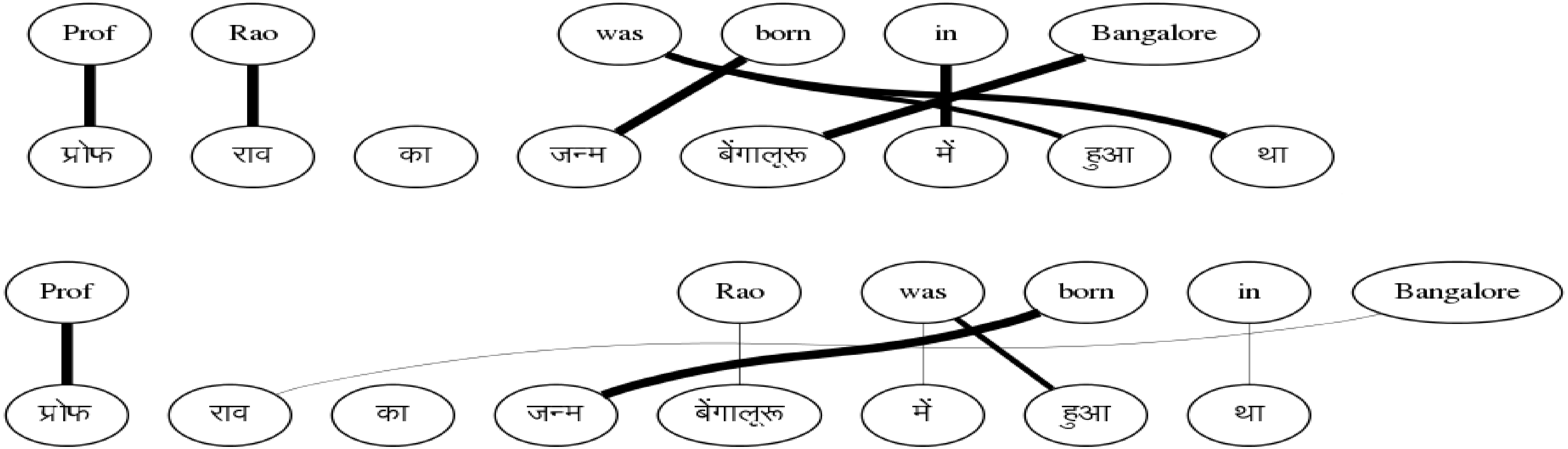
- *We don't know the best alignment*
- *So, we consider all alignments while estimating word translation probabilities*
- *Instead of taking only the best alignment, we consider all alignments and weigh the word alignments with the alignment probabilities*

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

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

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)

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 **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

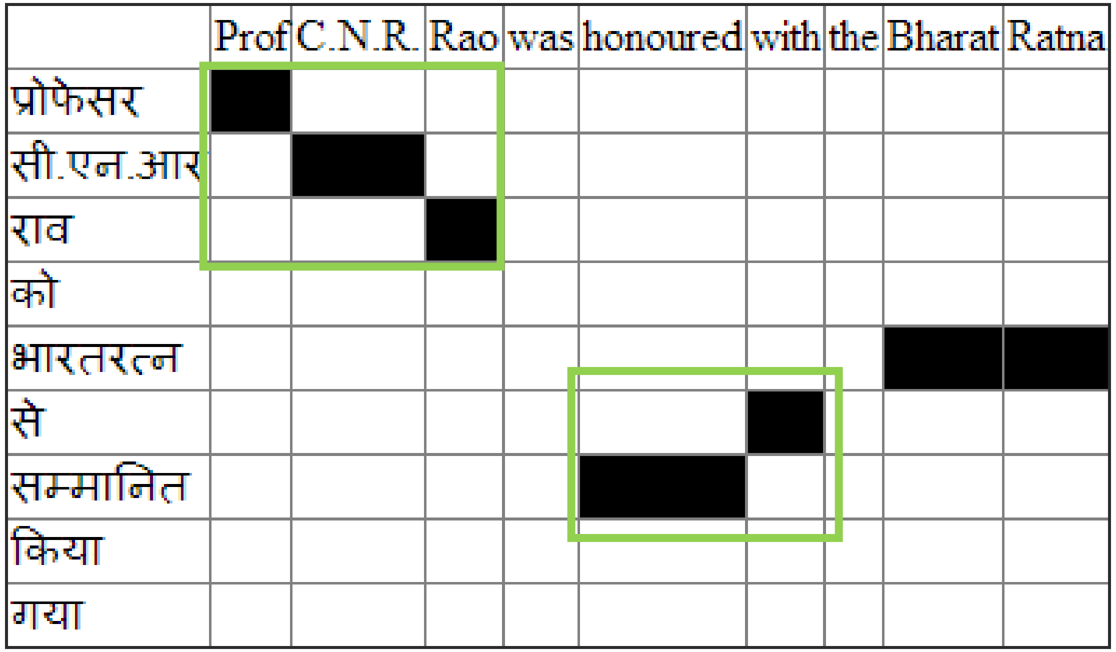
The Prime Minister of India	भारत के प्रधान मंत्री
is running fast	तेज भाग रहा है
the boy with the telescope	दूरबीन से लड़के को
Rahul lost the match	राहुल मुकाबला हार गया

- Learn the **Phrase Translation Probabilities**

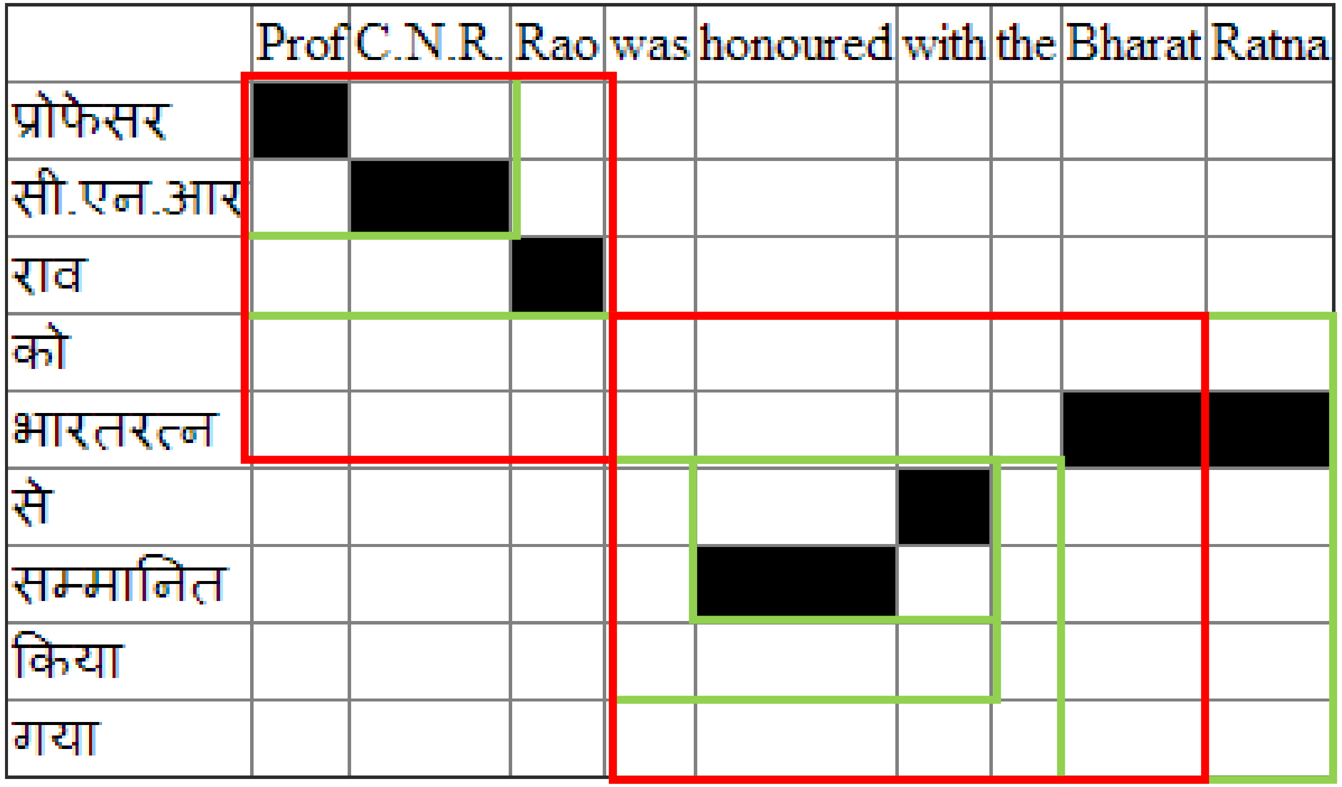
Prime Minister of India	भारत के प्रधान मंत्री India of Prime Minister	0.75
Prime Minister of India	भारत के भूतपूर्व प्रधान मंत्री India of former Prime Minister	0.02
Prime Minister of India	प्रधान मंत्री Prime Minister	0.23

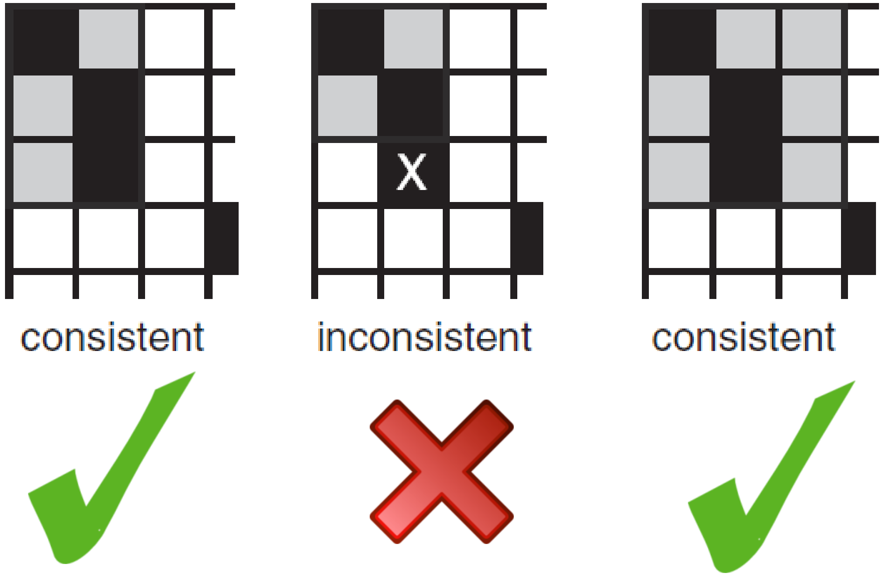
Learning Phrase Tables from Word Alignments

- 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

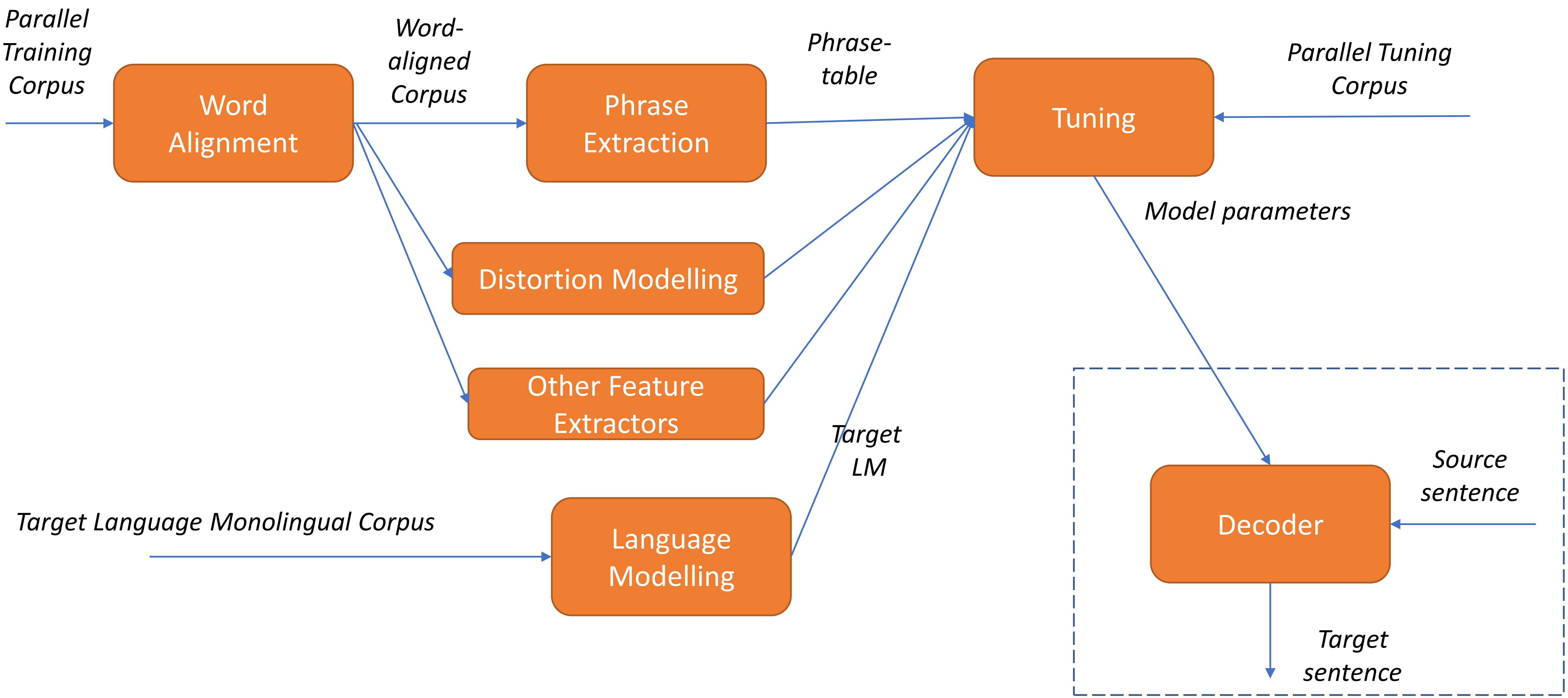
- Directly model the posterior probability p**(e|f)**
- Use the Maximum Entropy framework

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

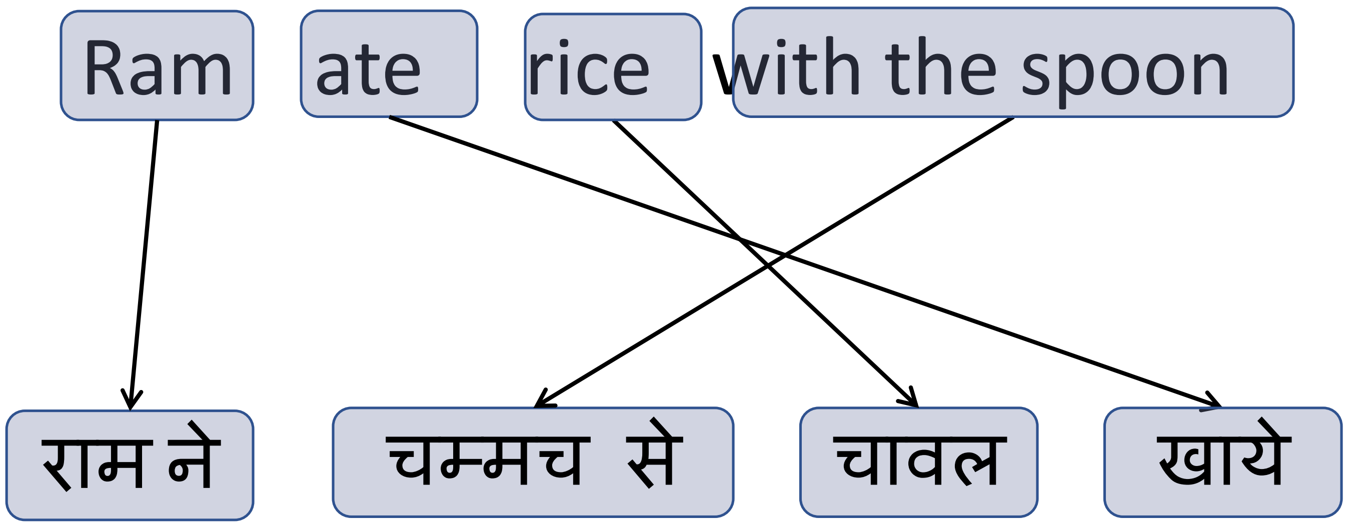
$$e^* = \arg\max_{e_i} \sum_i \lambda_i h_i(f_1^I, e_1^J)$$

  - 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



Decoding

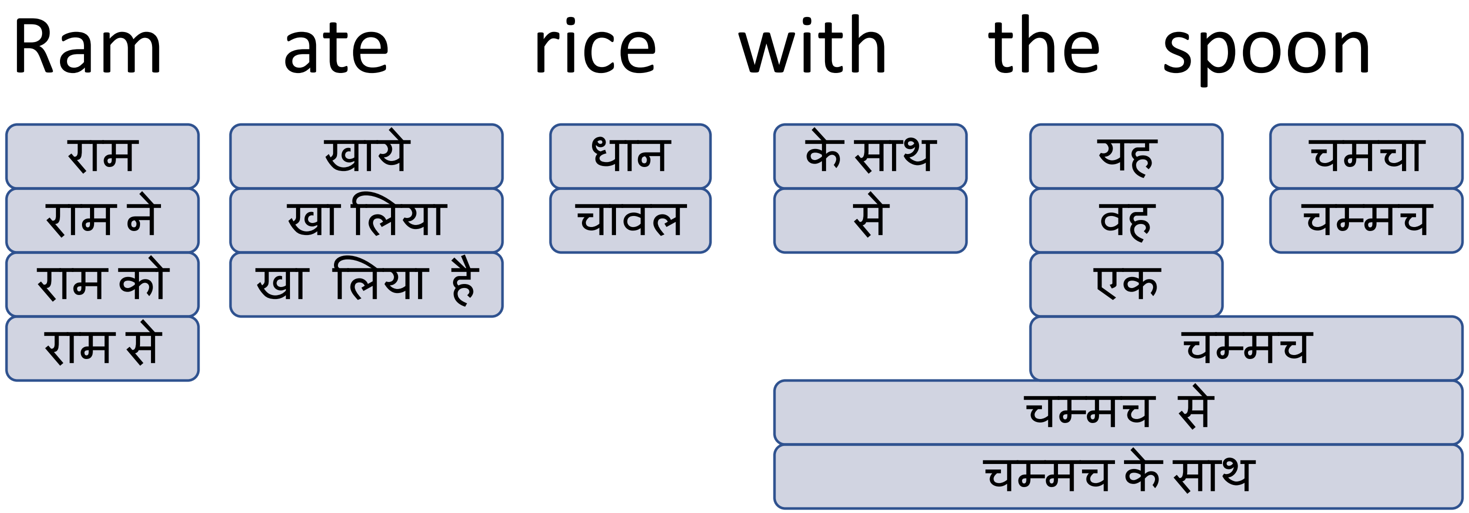


Searching for the best translations in the space of all translations

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

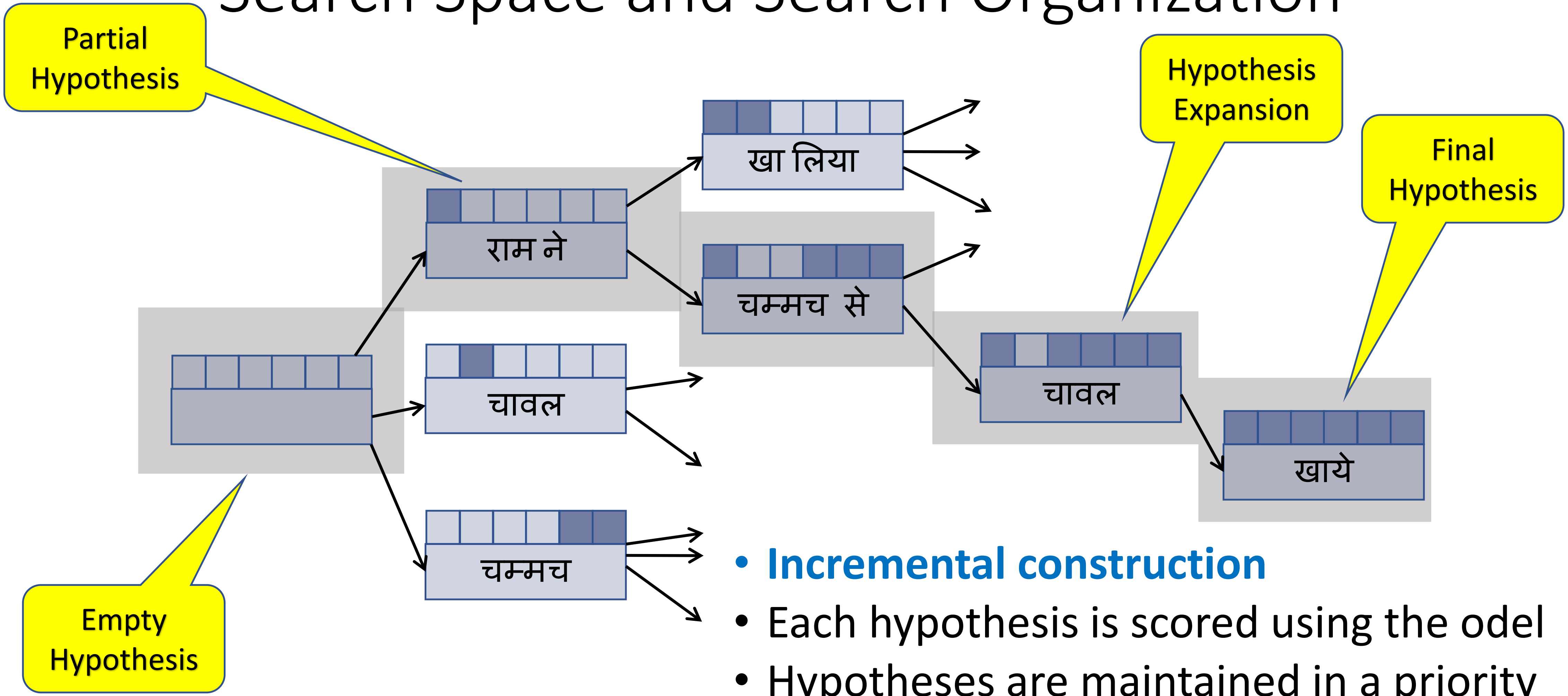
Decoding is challenging

- We picked the phrase translation that made sense to us
- The computer has less intuition
- Phrase table may give many options to translate the input sentence
- Multiple possible word orders



An <u>NP complete</u> search problem ➔ Needs a heuristic search method

Search Space and Search Organization



- **Incremental construction**
- Each hypothesis is scored using the odel
- Hypotheses are maintained in a priority queue
- Limit to the reordering window for efficiency

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

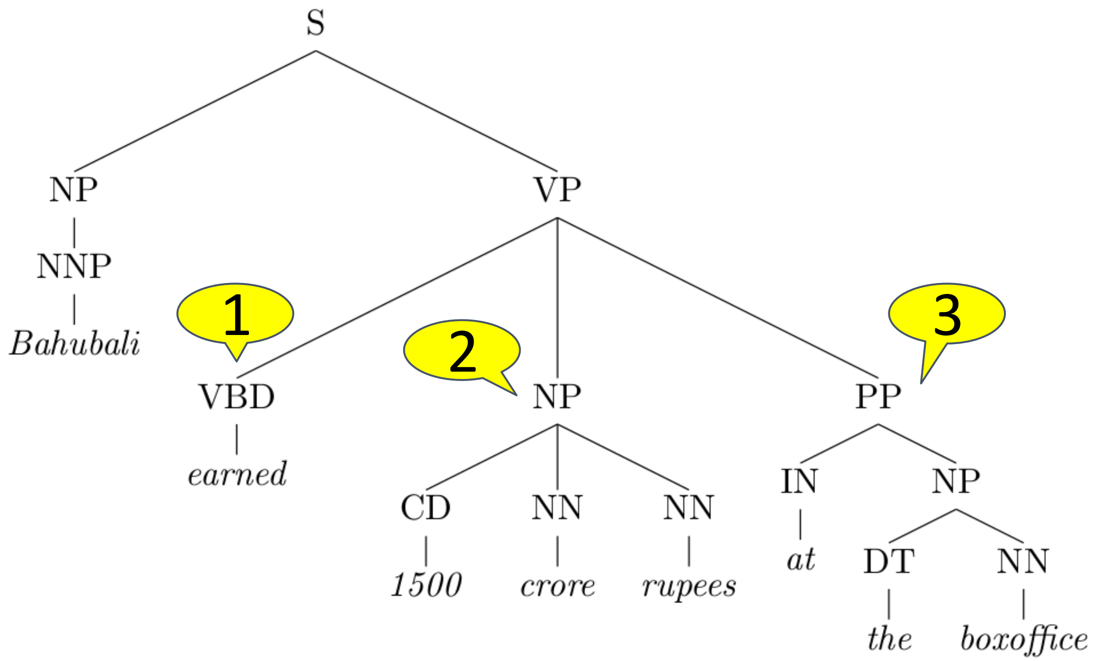
Solution: Let's help PB-SMT with some preprocessing of the input

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

Let's take an example

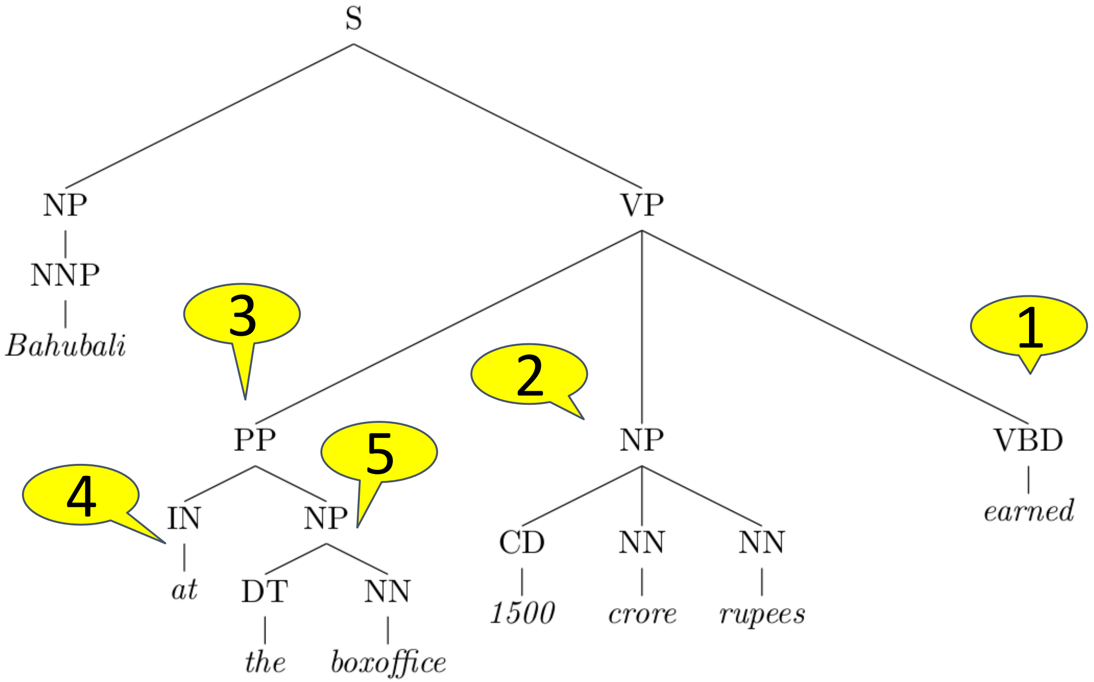
Bahubali earned more than 1500 crore rupee sat the boxoffice

Parse the sentence to understand its syntactic structure



Apply rules to transform the tree

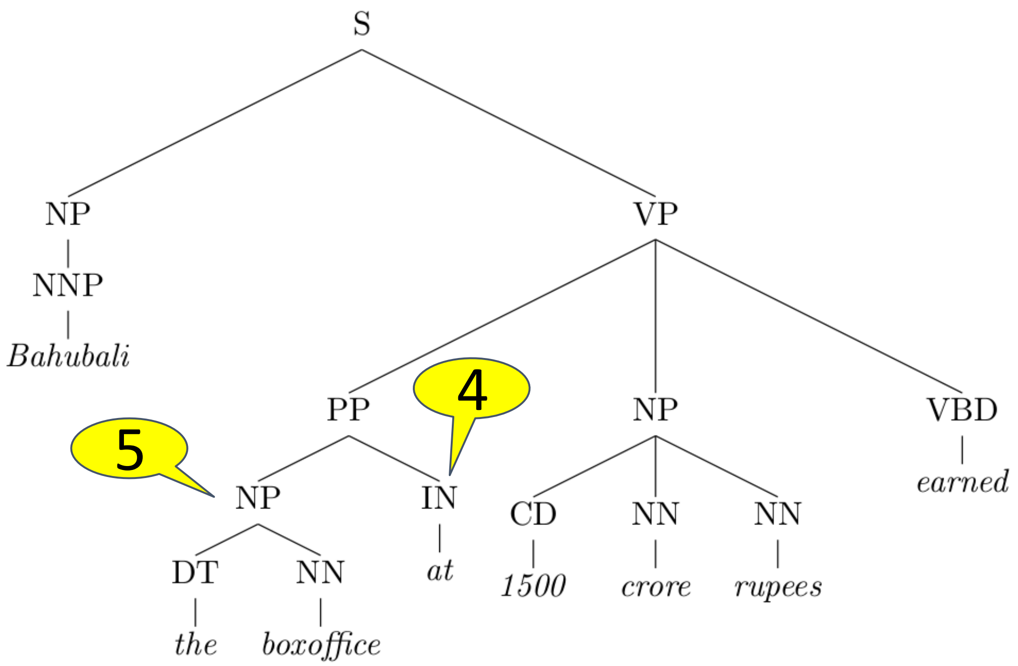
VP → VBD NP PP ⇒ VP → PP NP VBD



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

Prepositions in English become postpositions in Hindi

PP → 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*

$\Rightarrow$ *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

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

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

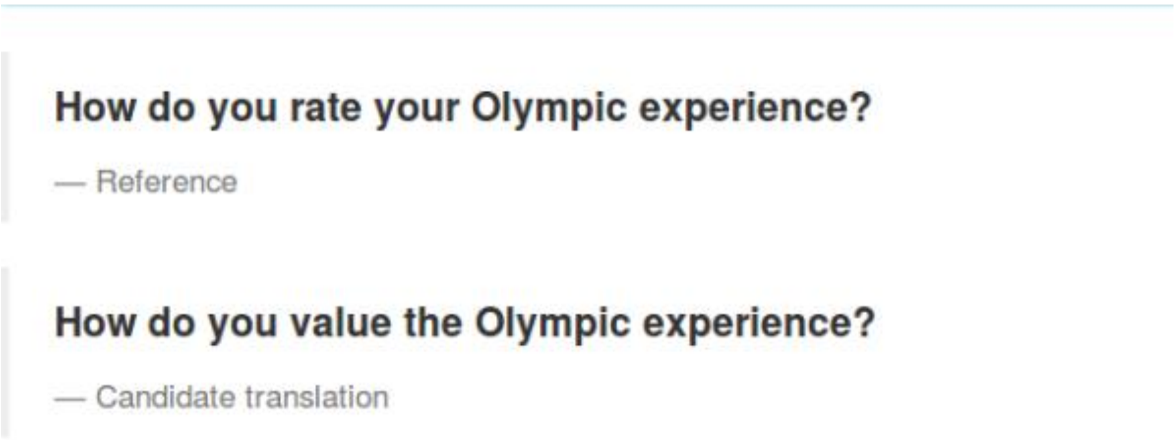
For example, I am attending a lecture

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?

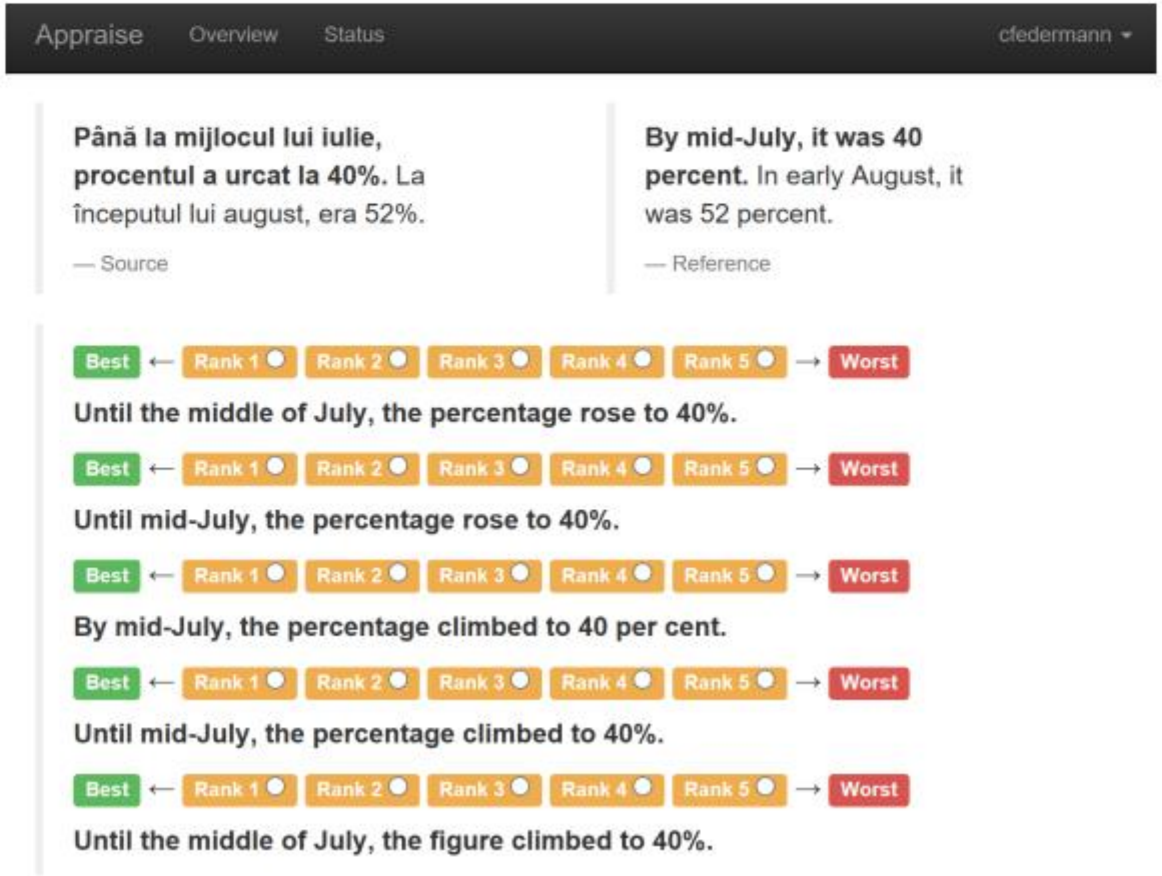
5 = All
4 = Most
3 = Much
2 = Little
1 = None

Fluency:

Is the sentence grammatically valid?

5 = Flawless
4 = Good
3 = Non-native
2 = Disfluent
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

Key idea of Automatic evaluation:

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

- 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

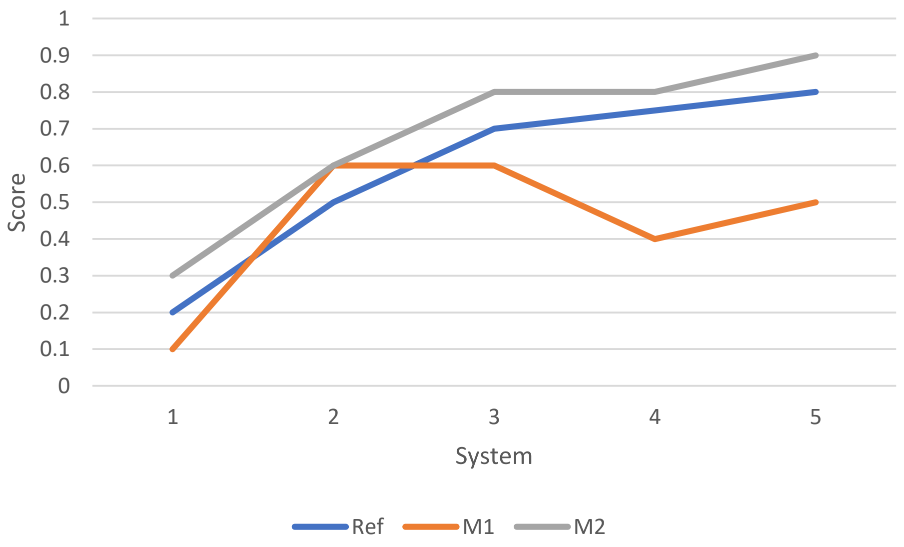
Multiple references are better

Some popular automatic evaluation metrics

- BLEU (Bilingual Evaluation Understudy)
- TER (Translation Edit Rate)
- METEOR (Metric for Evaluation of Translation with Explicit Ordering)

How good is an automatic metric?

How well does it correlate with human judgment?



Neural Machine Translation

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

home	0
water	1
house	2
tap	3

1	0	0	0

0	1	0	0

0	0	1	0

0	0	0	1

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

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

Every word is represented by a vector of numbers

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

- *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

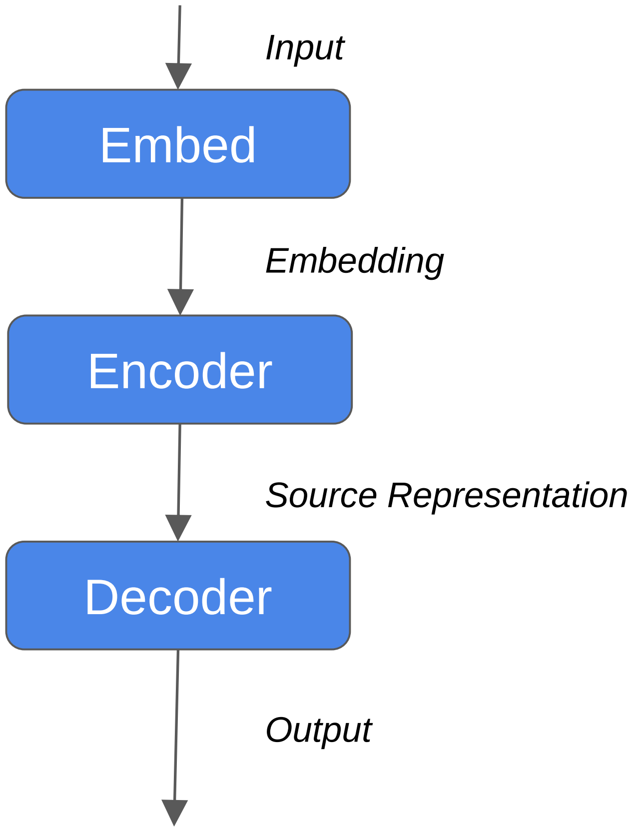
- *Vector similarity measures like cosine similarity*
- *Since* representations of `home` *and* `house`, *we may be able to translate* `house`

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



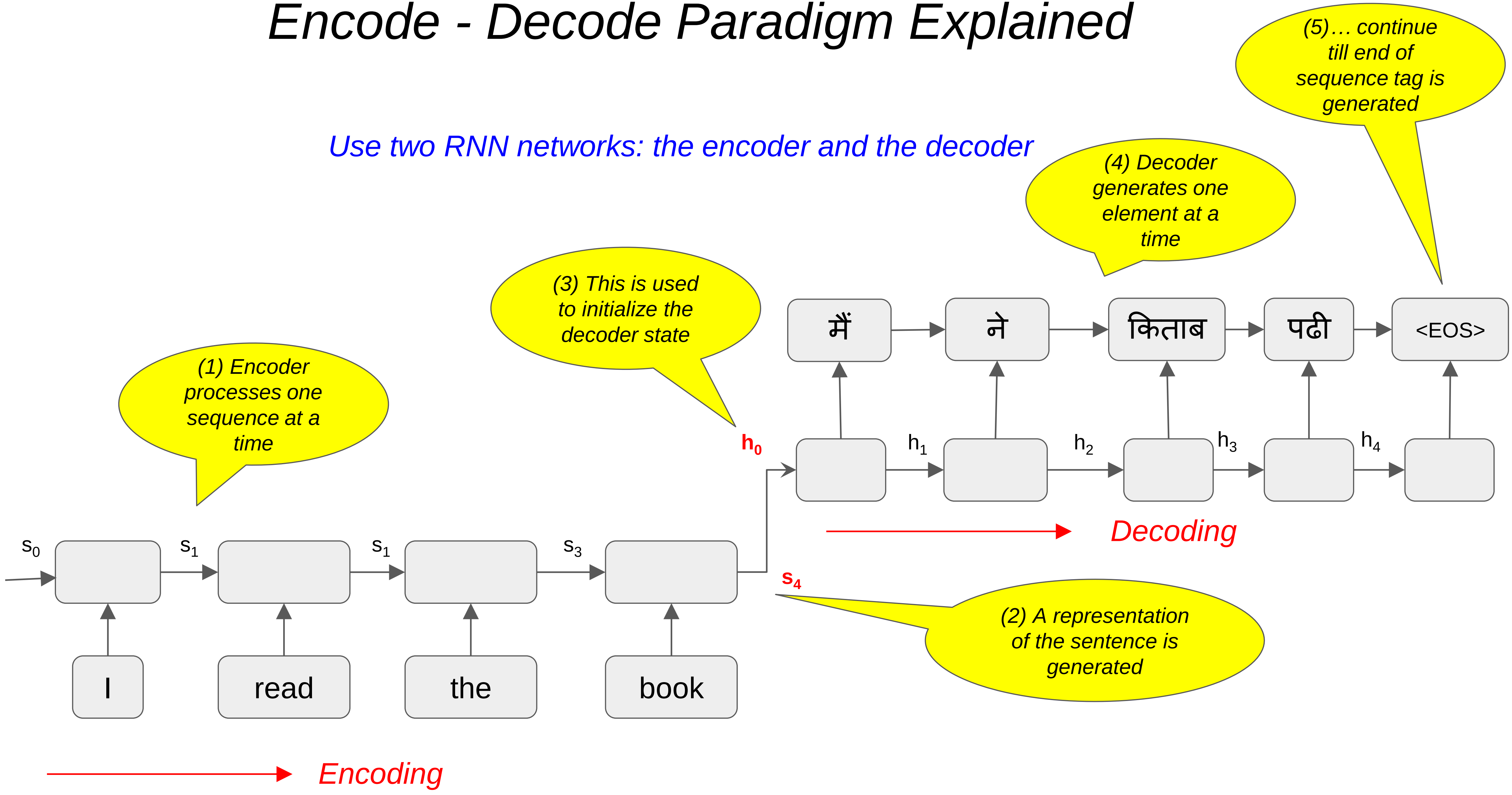
Entire input sequence is processed before generation starts
⇒ In PBSMT, generation was piecewise

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



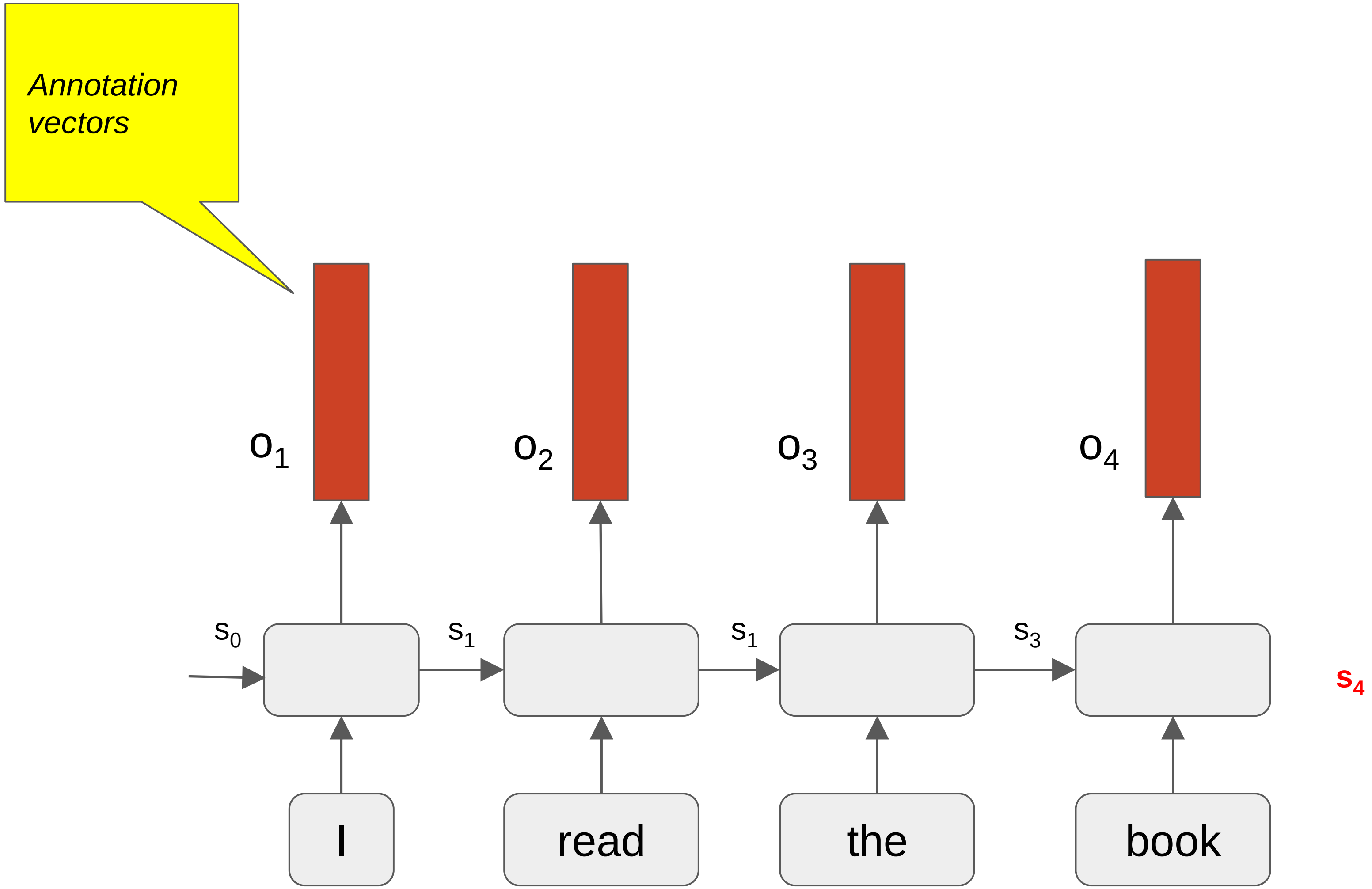
This approach reduces the entire sentence representation to a single vector

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***

How should the decoder use the set of annotation vectors while predicting the next character?

Key Insight:

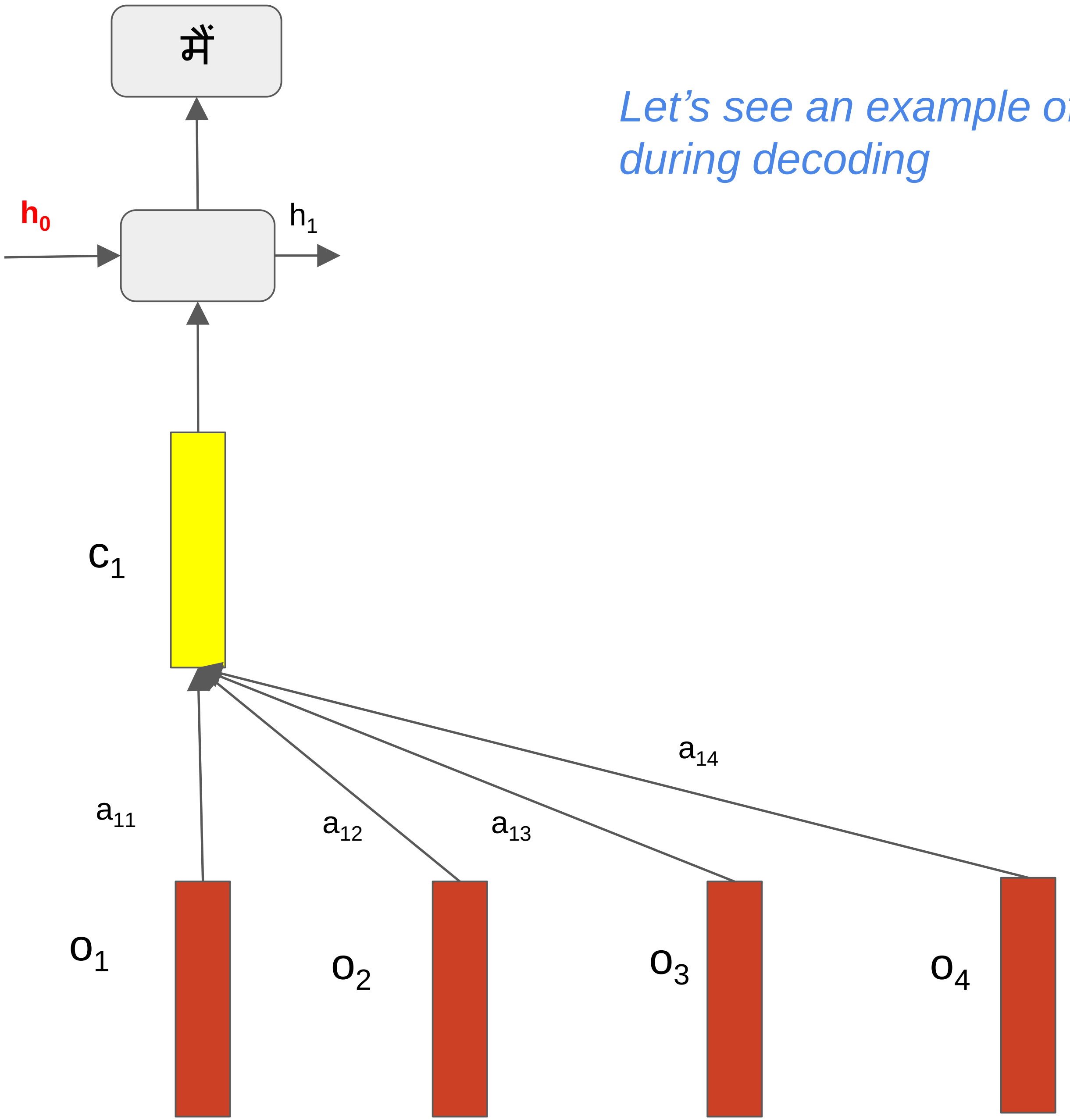
(1) Not all annotation vectors are equally important for prediction of the next element
(2) The annotation vector to use next depends on what has been generated so far by the decoder

eg. To generate the 3rd target word, the 3rd annotation vector (hence 3rd 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**

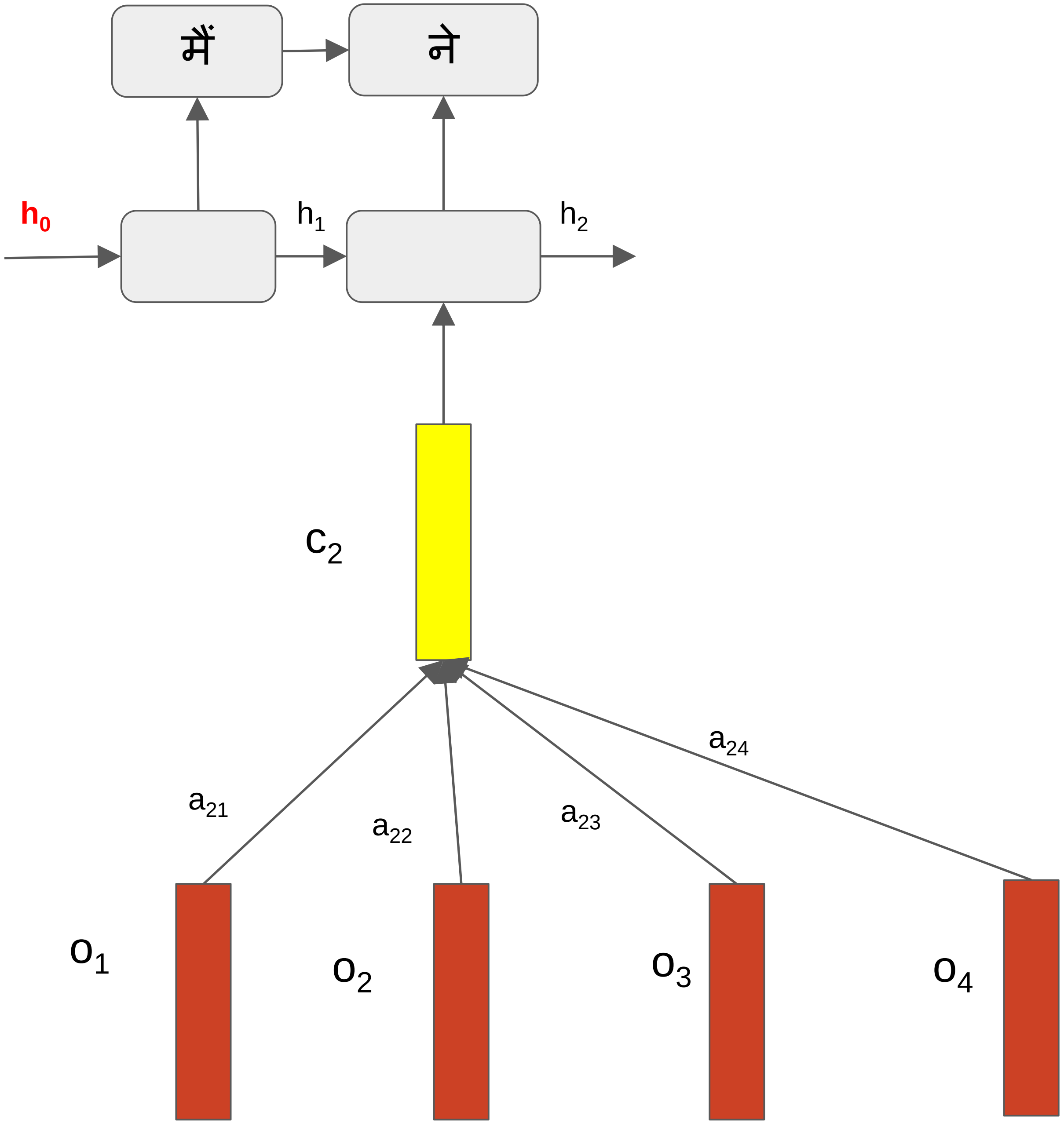
This averaged ***context vector*** is an input to the decoder

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

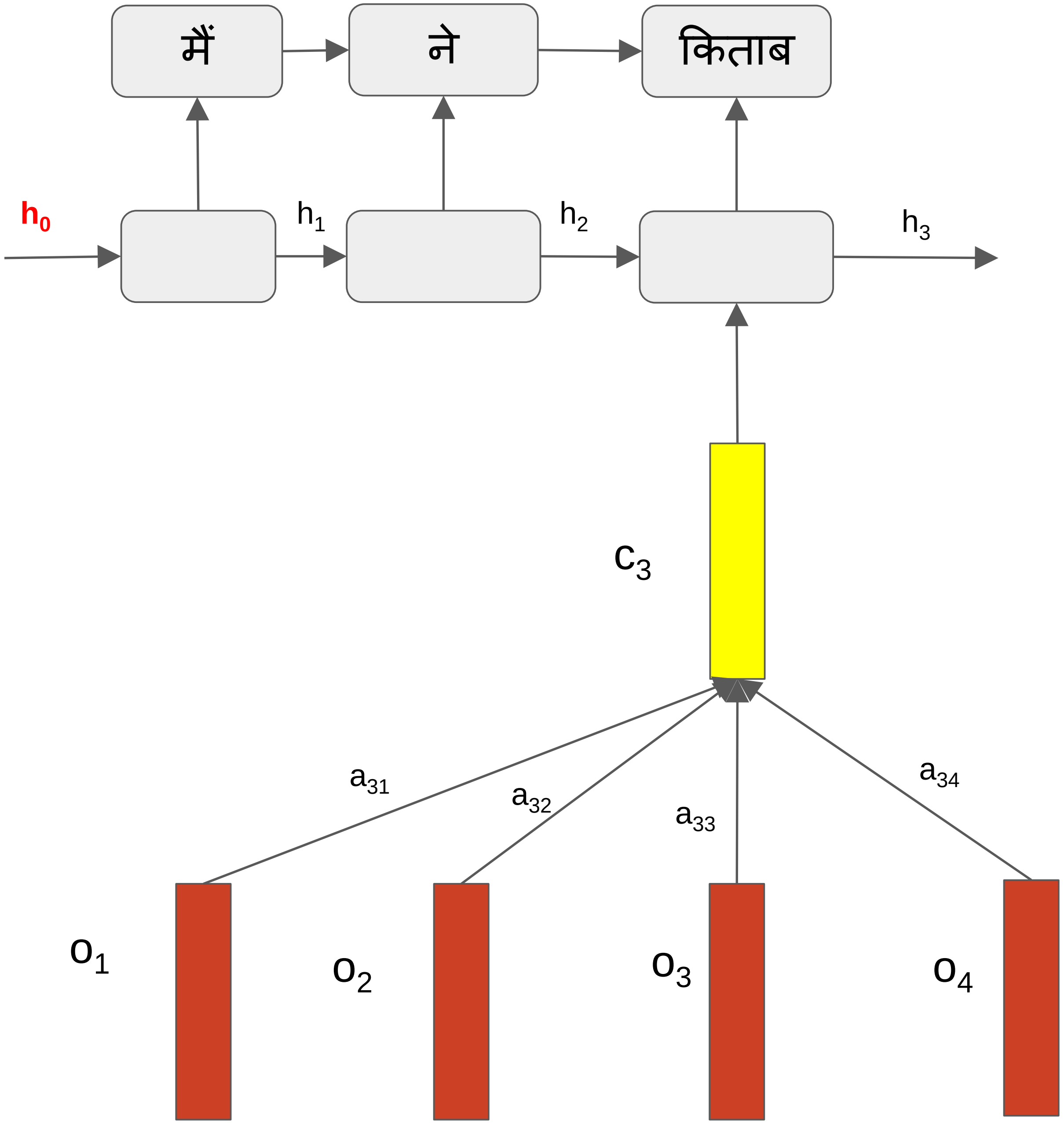


$$c_i = \sum_{j=1}^{n} a_{ij} o_j$$

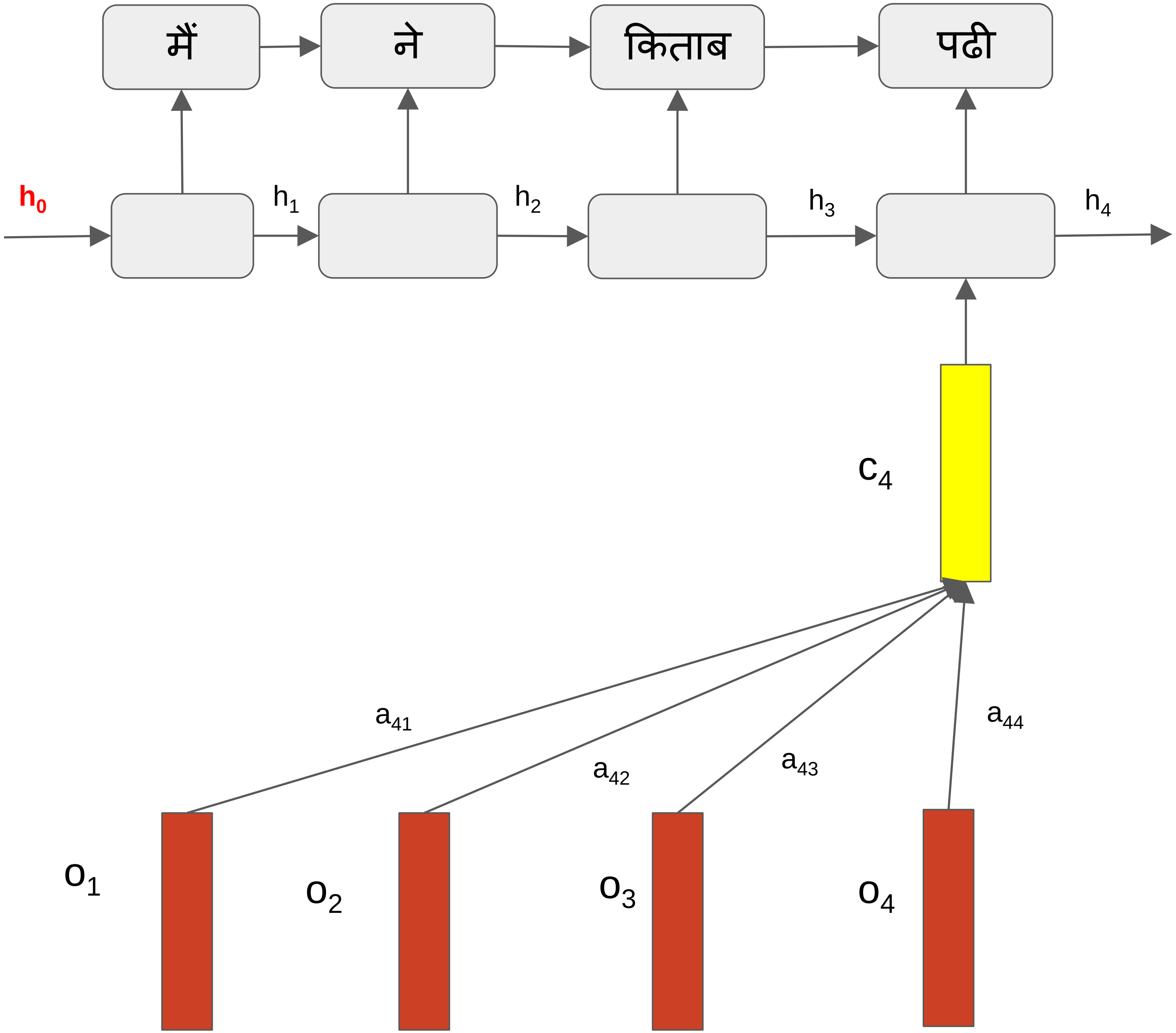
For generation of i^{th} output character:
c_i : context vector
a_{ij} : annotation weight for the j^{th} annotation vector
o_j: j^{th} annotation vector



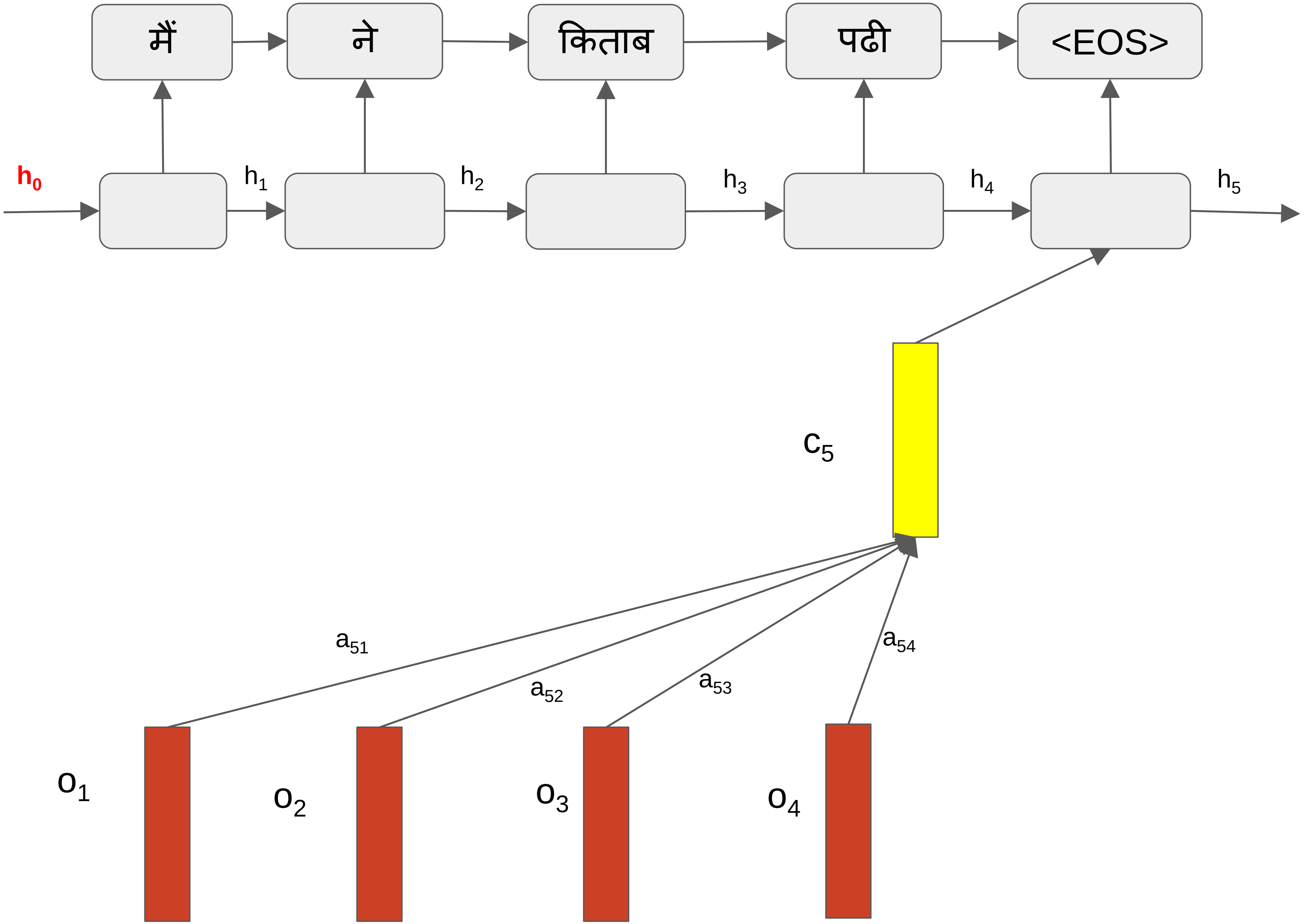
मैं
ने
h0
h1
h2
c2
a21
a22
a23
a24
o1
o2
o3
o4



मैं
ने
किताब
h0
h1
h2
h3
c3
a31
a32
a33
a34
o1
o2
o3
o4



मैं
ने
किताब
पढी
h0
h1
h2
h3
h4
c4
a41
a42
a43
a44
o1
o2
o3
o4


मैं
ने
किताब
पढी
<EOS>
h_0
h_1
h_2
h_3
h_4
h_5
c_5
a_{51}
a_{52}
a_{53}
a_{54}
o_1
o_2
o_3
o_4

But we do not know the attention weights?
How do we find them?

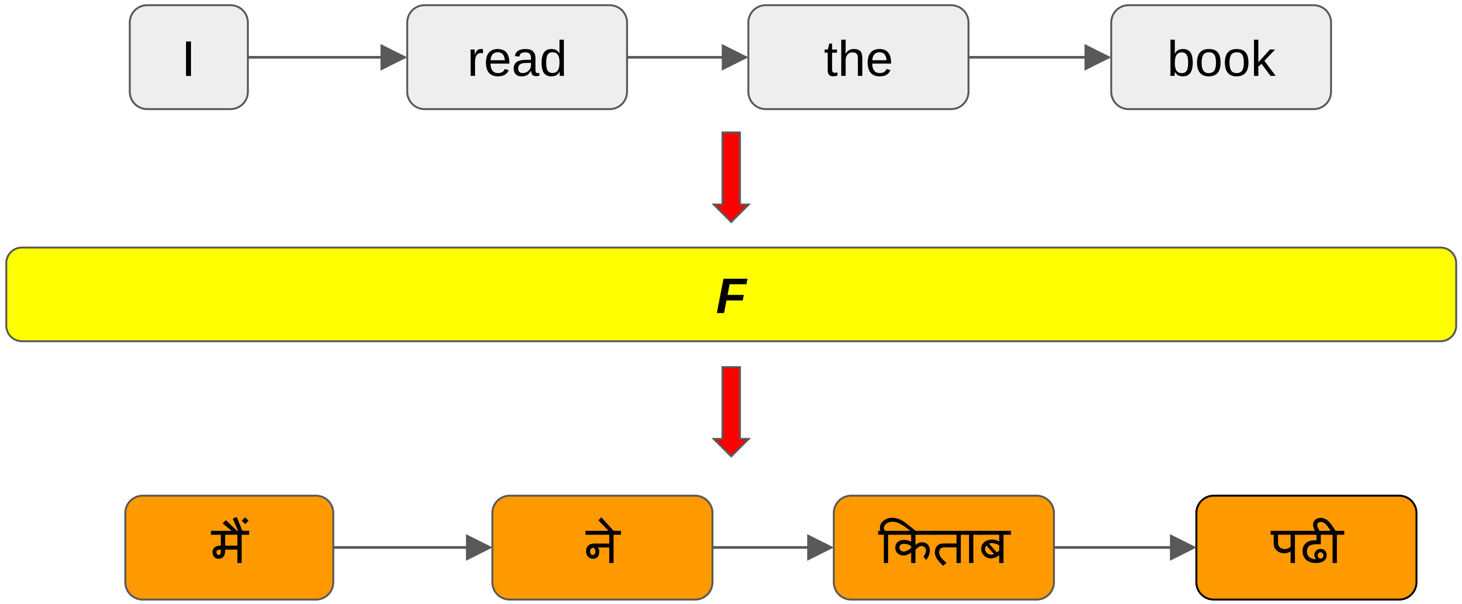
Let the training data help you decide!!

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*
- *End-to-end training*

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 ⇒ Summary*
- *Question answering: Question ⇒ Answer*
- *Image labelling: Image ⇒ Label*
- *Transliteration: character sequence ⇒ character sequence*

Machine Translation for Related Languages

Related Languages

Related by Genealogy

Related by Contact

Language Families

Dravidian, Indo-European, Turkic

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

Linguistic Areas

Indian Subcontinent, Standard Average European

(Trubetzkoy, 1923)

Related languages may not belong to the same language family!

Key Similarities between related languages

भारताच्या स्वातंत्र्यदिनानिमित्त अमेरिकेतील लॉस एन्जल्स शहरात कार्यक्रम आयोजित करण्यात आला
bhAratAcyA svAta.ntryadinAnimitta ameriketIla lOsa enjalsa shaharAta kAryakrama Ayojita karaNyAta AlA

Marathi

भारता च्या स्वातंत्र्य दिना निमित्त अमेरिके तील लॉस एन्जल्स शहरा त कार्यक्रम आयोजित करण्यात आला
bhAratA cyA svAta.ntrya dinA nimitta amerike tIla lOsa enjalsa shaharA ta kAryakrama Ayojita karaNyAta AlA

Marathi segmented

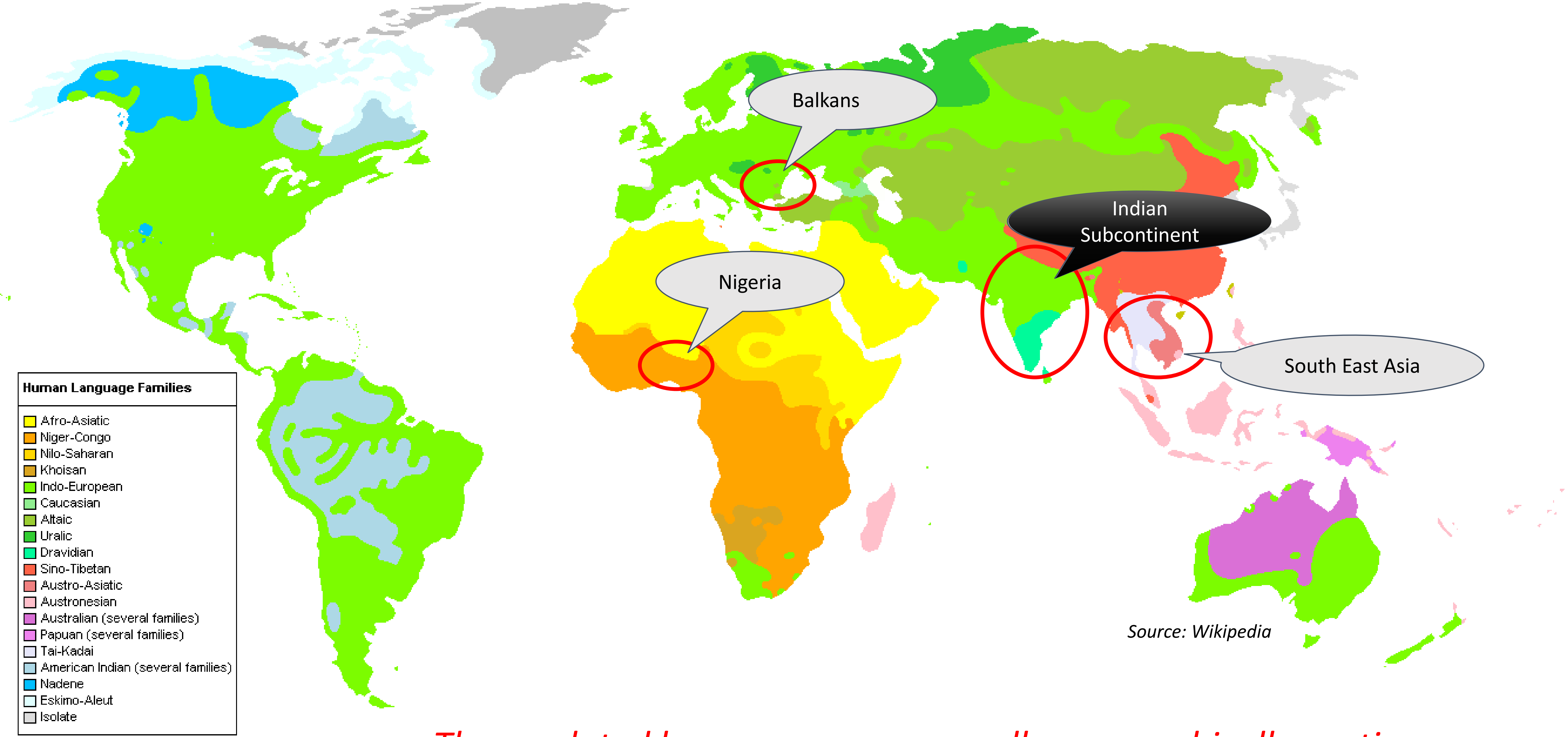
भारत के स्वतंत्रता दिवस के अवसर पर अमरीका के लॉस एन्जल्स शहर में कार्यक्रम आयोजित किया गया
bhArata ke svata.ntratA divasa ke avasara para amarIkA ke losa enjalsa shahara me.n kAryakrama Ayojita kiyA gayA

Hindi

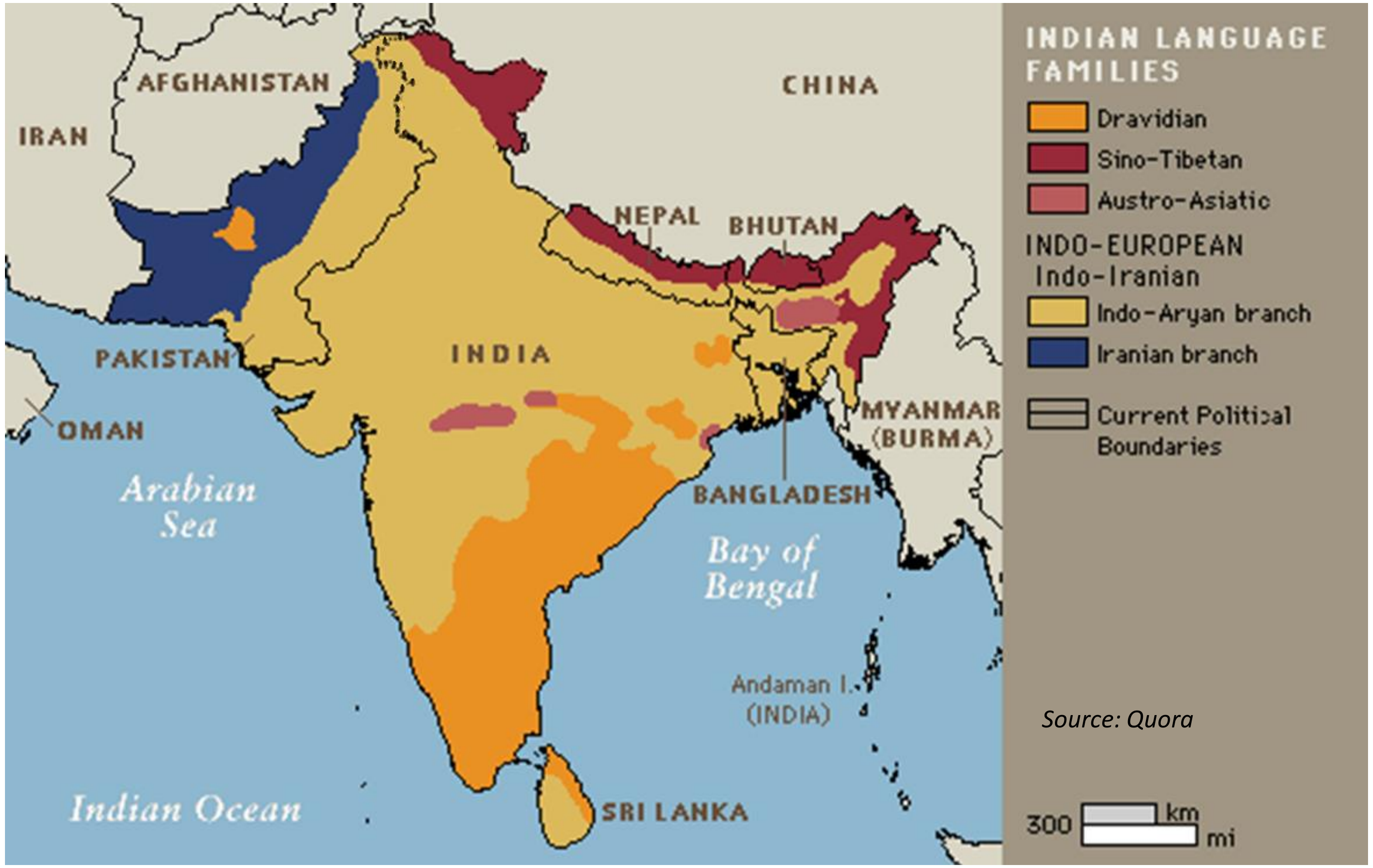
Lexical: share significant vocabulary (cognates & loanwords)

Morphological: correspondence between suffixes/post-positions

Syntactic: share the same basic word order



These related languages are generally geographically contiguous



- 5 language families (+ 2 to 3 on the Andaman & Nicobar Islands)
- 22 scheduled languages
- 11 languages with more than 25 million speakers
- Highly multilingual country

Naturally, lot of communication between such languages (government, social, business needs)

⇓

Most translation requirements also involves related languages

Between related languages

Hindi-Malayalam

Marathi-Bengali

Czech-Slovak

Related languages ⇐⇒ Link languages

Kannada,Gujarati ⇒ English

English ⇒ Tamil,Telugu

We want to be able to handle a large number of such languages

e.g. 30+ languages with a speaker population of 1 million + in the Indian subcontinent

Aren't "*language independent*" Statistical/Neural Machine Translation methods sufficient?

- Implicit assumptions increase need for:

  (1) Parallel Corpora (2) Linguistic Resources (3) Language specific processing

- *'Limited language independence'* can be achieved between some languages if we can make assumptions that hold across all these languages

- Related languages can serve as a good level of abstraction to utilize linguistic regularities:
  - Reduce parallel corpora
  - Reduce linguistic resource requirements
  - Better Generalization

Utilizing Lexical Similarity

Lexically Similar Languages
*(Many words having similar **form** and **meaning**)*

- Cognates

a common etymological origin

roTI (hi)	*roTIA (pa)*	*bread*
bhai (hi)	*bhAU (mr)*	*brother*

- Named Entities

do not change across languages

mu.mbaI (hi)	mu.mbaI (pa)	mu.mbaI (pa)
keral (hi)	k.eraLA (ml)	keraL (mr)

- Loan Words

borrowed without translation

matsya (sa)	*matsyalu (te)*	*fish*
pazha.m (ta)	*phala (hi)*	*fruit*

- Fixed Expressions/Idioms

MWE with non-compositional semantics

dAla me.n kuCha kAlA honA	*(hi)*	*Something fishy*
dALa mA kAlka kALu hovu	(gu)	

We will find more matches at the sub-word level

Can we use subwords as translation units?

Which subword should we use?

W: राजू , घराबाहेर जाऊ नको .

O: रा जू _ , _ घ रा बा हे र _ जा ऊ _ न को _ .

Simple Units of Text Representation

Transliterate unknown words *[Durrani, etal. (2010), Nakov & Tiedemann (2012)]*

(a) Primarily used to handle proper nouns

(b) Limited use of lexical similarity

स्वातंत्र्य → स्वतंत्रता

→ *Translation of shared lexically similar words can be seen as kind of transliteration*

Character *[Vilar, etal. (2007), Tiedemann (2009)]*

Limited context of character level representation

Character n-gram ⇒ increase in data sparsity

→ *Limited benefit ….*

… just for closely related languages

Macedonian - Bulgarian, Hindi-Punjabi, etc.

Orthographic Syllable *(Kunchukuttan & Bhattacharyya, 2016a)*

(CONSONANT) + VOWEL

Examples: ca, cae, coo, cra, की (kI), प्रे (pre)

अभिमान ➔ अ भि मा न

<u>Pseudo-Syllable</u>

True Syllable ⇒ Onset, Nucleus and Coda

Orthographic Syllable ⇒ Onset, Nucleus

- Generalization of ***akshara***, the fundamental organizing principle of Indian scripts
- Linguistically motivated, variable length unit
- *Number of syllables in a language is finite*

Byte Pair Encoded (BPE) Unit

(Kunchukuttan & Bhattacharyya, 2017a; Nguyen and Chang, 2017)

- *There may be frequent subsequences in text other than syllables*
- *These subsequences may not be valid linguistic units*
- *But they represent statistically important patterns in text*

How do we identify such frequent patterns?

Byte Pair Encoding (Sennrich et al, 2016), wordpieces (Wu et al, 2016),

Huffman encoding based units (Chitnis & DeNero, 2015)

Byte Pair Encoded (BPE) Unit

Byte Pair Encoding is a 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

<u>*Words to encode*</u> <u>*Iterations*</u>

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

<u>Data-dependent segmentation</u>

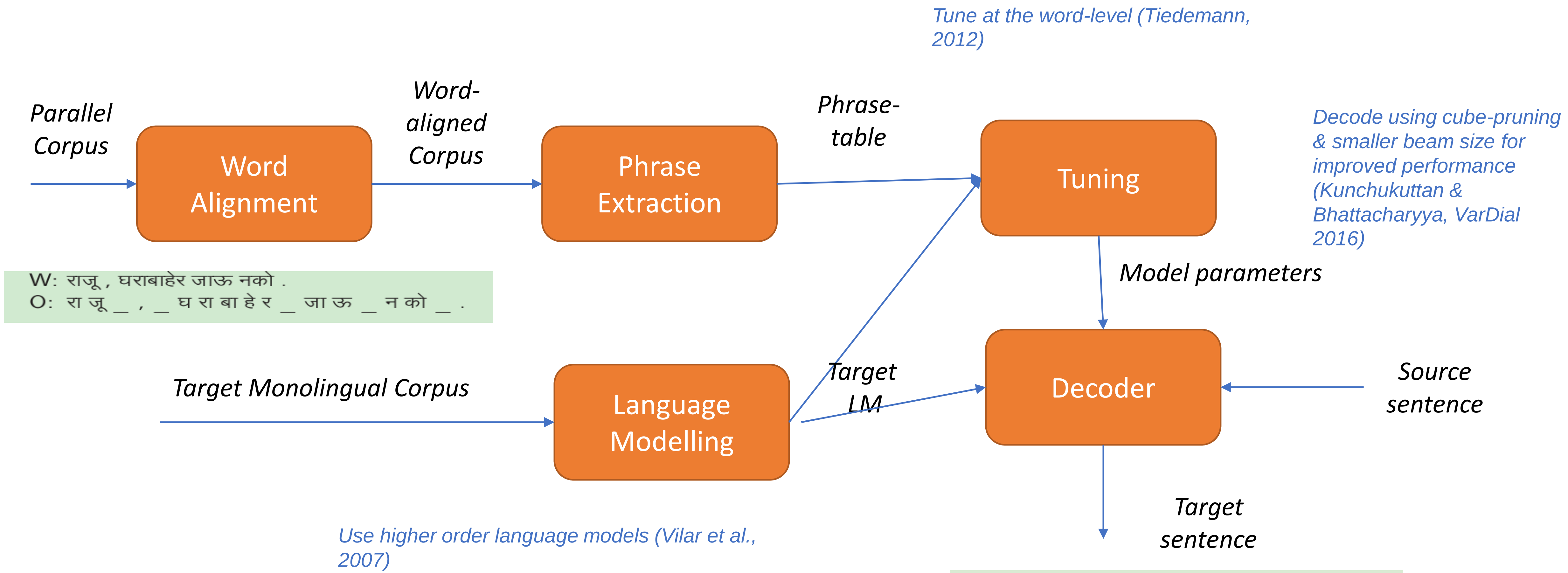
- Inspired from compression theory
- MDL Principle *(Rissansen, 1978)* ⇒ Select segmentation which maximizes data likelihood

Example of various translation units

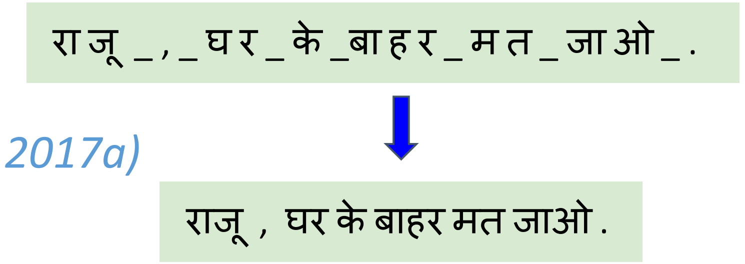
Basic Unit	Symbol	Example	Transliteration
Word	W	घरासमोरचा	gharAsamoracA
Morph Segment	M	घरा समोर चा	gharA samora cA
Orthographic Syllable	O	घ रा स मो र चा	gha rA sa mo racA
Character unigram	C	घ र ◌ा स म ◌ो र च ◌ा	gha r A sa m o ra c A
something that is in front of home:	ghara=home, samora=front, cA=of		

Various translation units for a Marathi word

Adapting SMT for subword-level translation



Significant improvement over level baselines (11-14%)



(Kunchukuttan & Bhattacharyya, 2016a; Kunchukuttan & Bhattacharyya, 2017a)

NMT with related languages on source side

(Nguyen and Chang, 2017)

We want Marathi → English translation ➔ but little parallel corpus is available
We have lot of Hindi → English parallel corpus

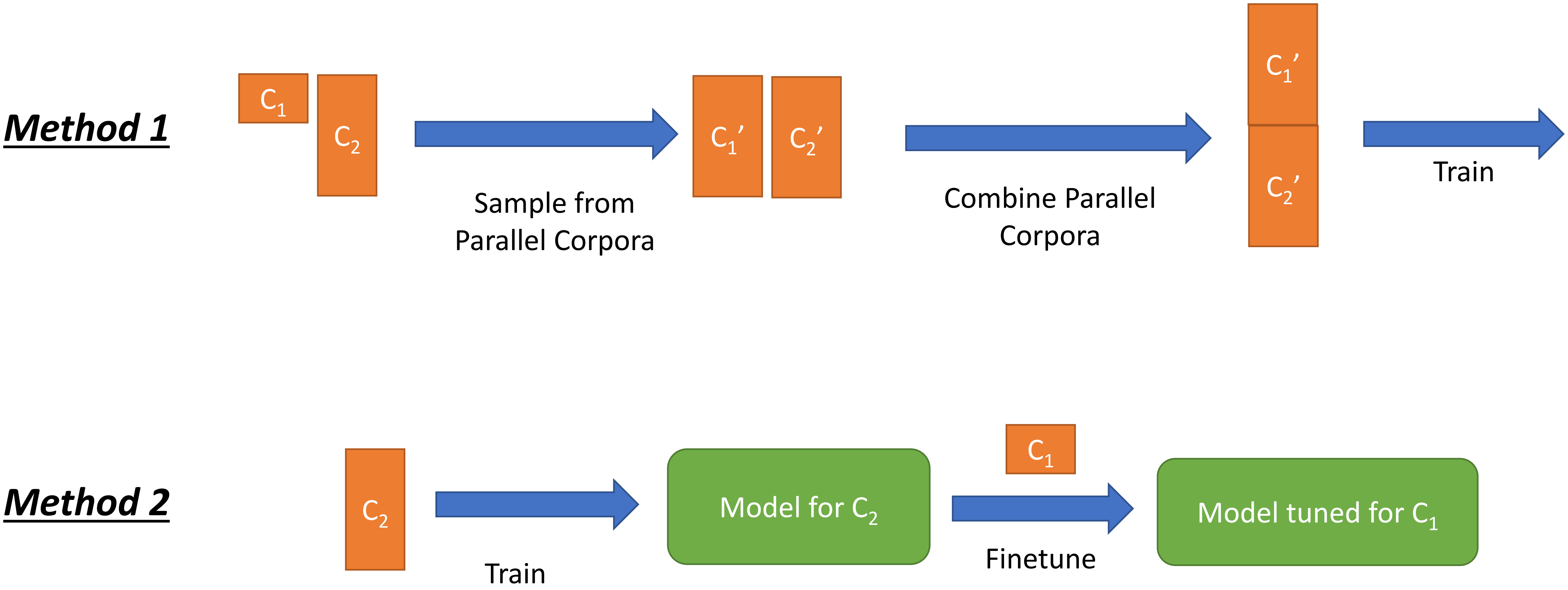
It is cold in Pune	पुण्यात थंड आहे
My home is near the market	माझा घर बाजाराजवळ आहे

I am going home	मैं घर जा रहा हूँ
It rained last week	पिछले हफ्ते बारिश हुई

Concat Corpora

It is cold in Pune	पुण्यात थंड आहे
My home is near the market	माझा घर बाजाराजवळ आहे
I am going home	मैं घर जा रहा हूँ
It rained last week	पिछले हफ्ते बारिश हुई

Training Multilingual NMT systems



What if the related languages use different scripts?

(Nguyen and Chang, 2017)

We want Gujarati → English translation ➔ but little parallel corpus is available
We have lot of Marathi → English parallel corpus

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

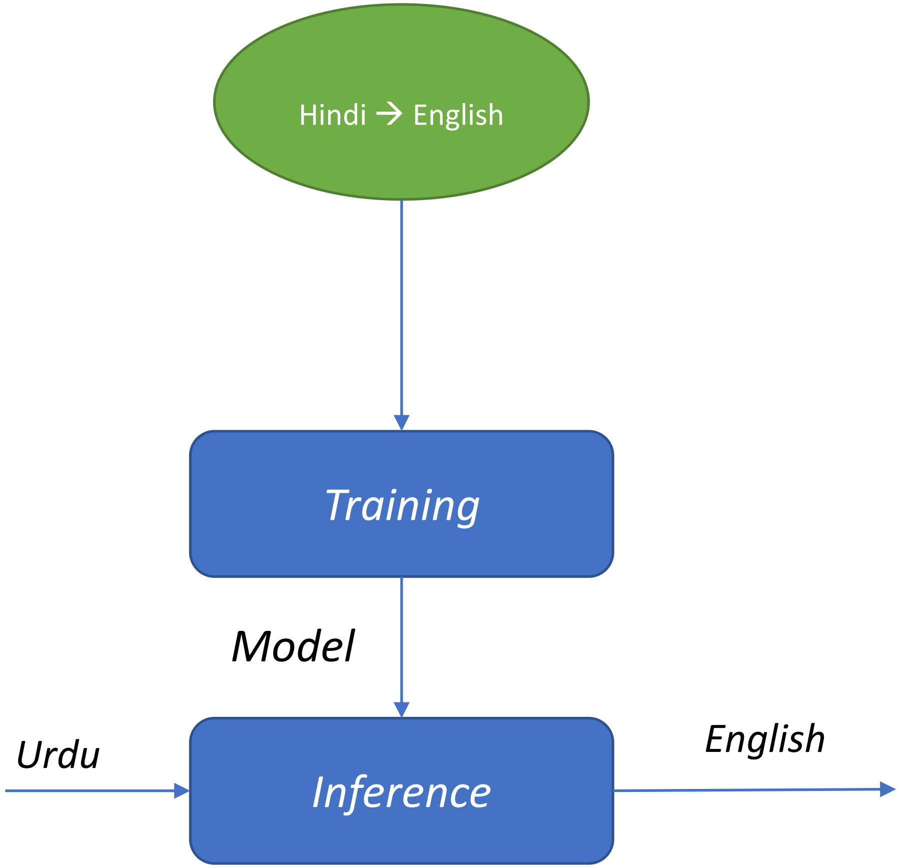
It is cold in Pune	पुण्यात थंड आहे
My home is near the market	माझा घर बाजाराजवळ आहे

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

Zeroshot Translation



How do we support multiple target languages with a single decoder?

A simple trick!

Append input with special token indicating the target language

For English-Hindi Translation

Original Input: *France and Croatia will play the final on Sunday*

Modified Input: *France and Croatia will play the final on Sunday* <hin>

Still an open problem

Utilizing Syntactic Similarity

(Kunchukuttan et al., 2014)

Phrase based MT is not good at learning word ordering

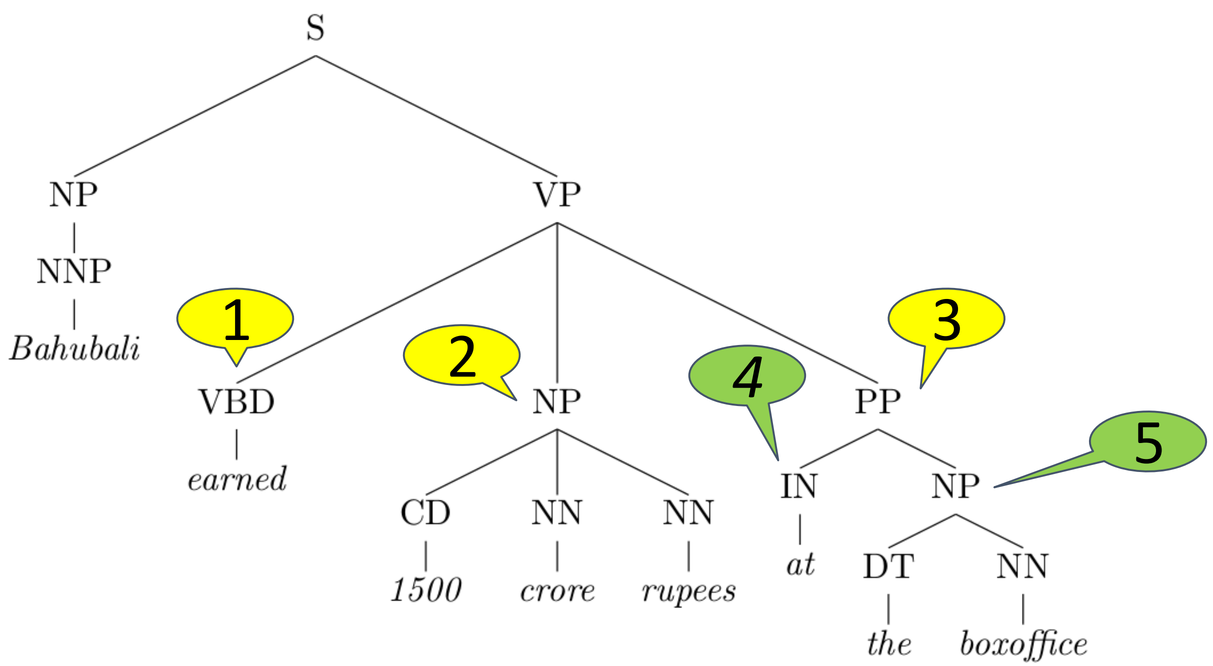
Solution: Let's help PB-SMT with some preprocessing of the input

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

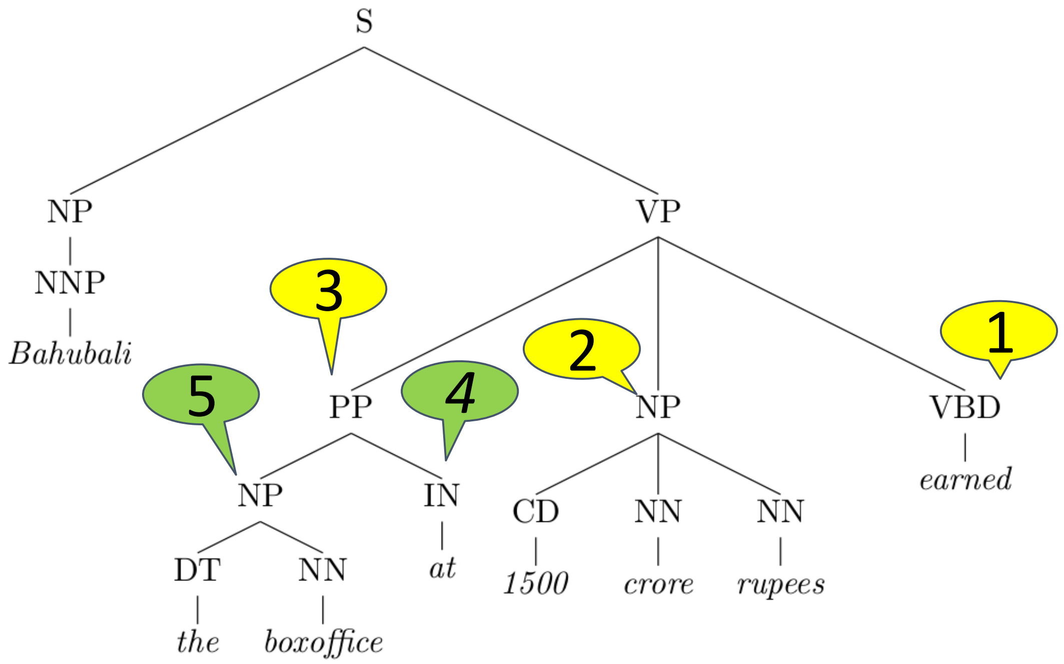
Parse the sentence to understand its syntactic structure



Apply rules to transform the tree

VP → VBD NP PP ⇒ VP → PP NP VBD

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

Can we reuse English-Hindi rules for English-Indian languages?

All Indian languages have the same basic word order

	Indo-Aryan						Dravidian		
	pan	**hin**	**guj**	**ben**	**mar**	**kok**	**tel**	**tam**	**mal**
Baseline	15.83	21.98	15.80	12.95	10.59	11.07	7.70	6.53	3.91
Generic	17.06	23.70	16.49	13.61	11.05	11.76	7.84	6.82	**4.05**
Hindi-tuned	**17.96**	**24.45**	**17.38**	**13.99**	**11.77**	**12.37**	**8.16**	**7.08**	4.02

(Kunchukuttan et al., 2014)

Generic reordering *(Ramanathan et al 2008)*

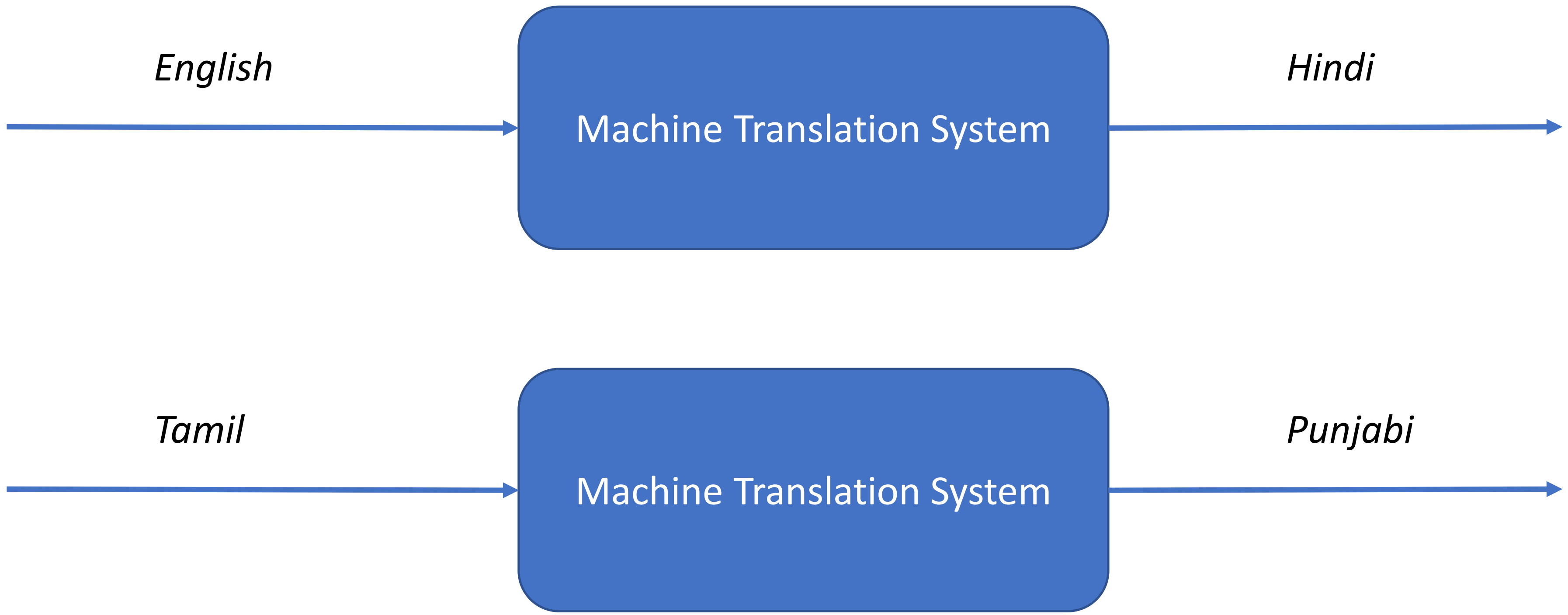
Basic reordering transformation for English→ Indian language translation

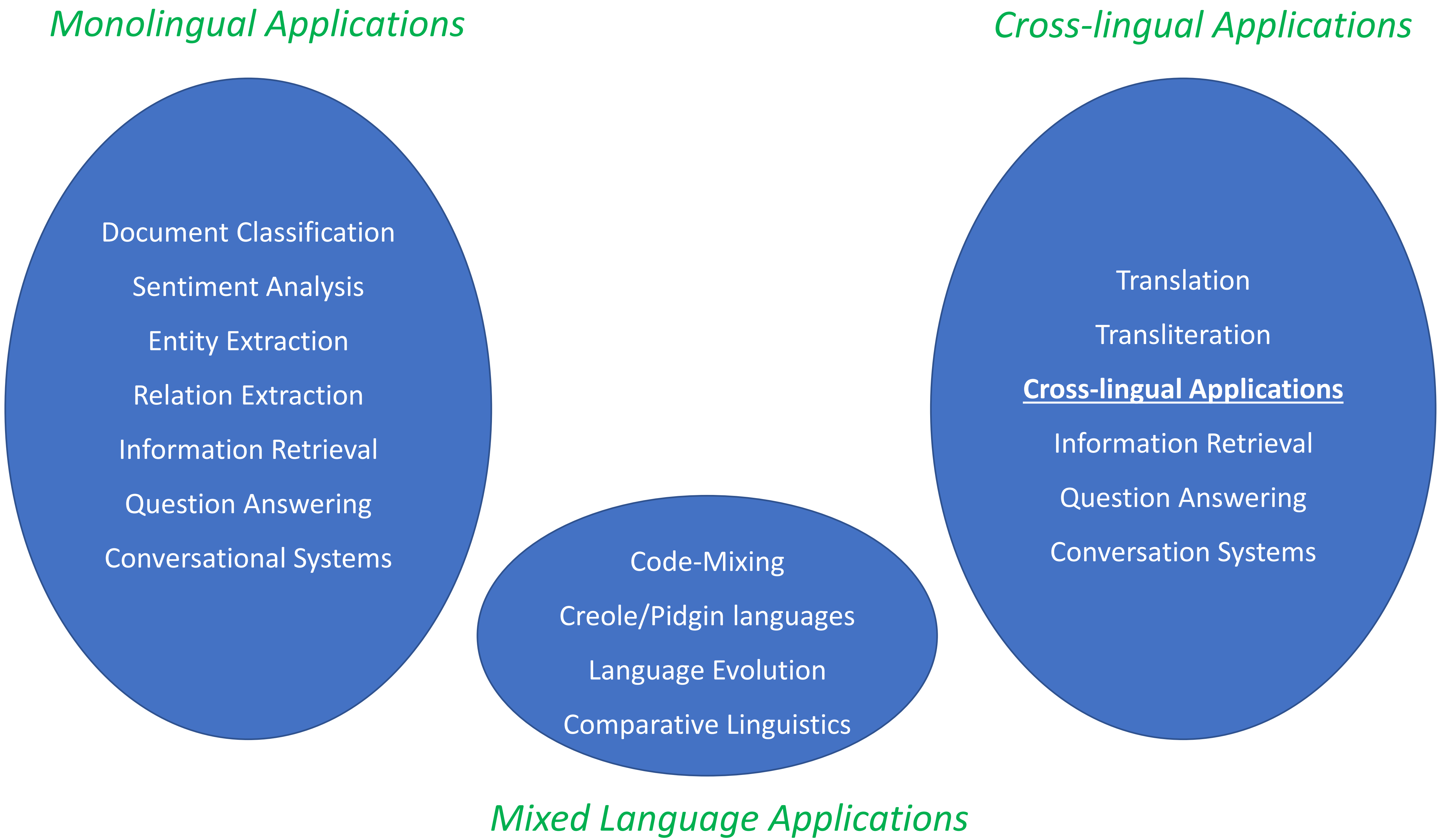
Hindi-tuned reordering *(Patel et al 2013)*

Improvement over the basic rules by analyzing English → Hindi translation output

Multilingual Learning

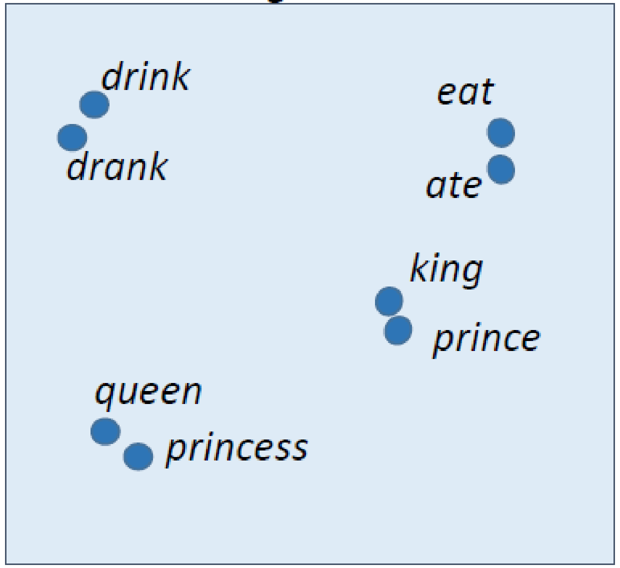
Broad Goal: Build NLP Applications that can work on different languages



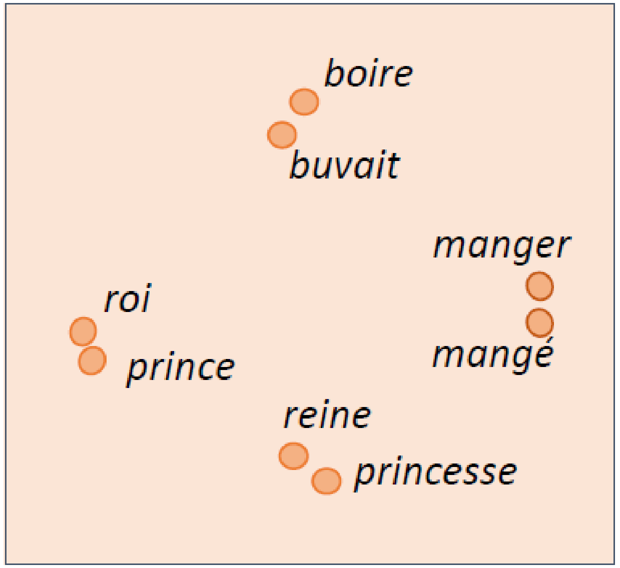

Monolingual Applications
Document Classification
Sentiment Analysis
Entity Extraction
Relation Extraction
Information Retrieval
Question Answering
Conversational Systems
Cross-lingual Applications
Translation
Transliteration
Cross-lingual Applications
Information Retrieval
Question Answering
Conversation Systems
Code-Mixing
Creole/Pidgin languages
Language Evolution
Comparative Linguistics
Mixed Language Applications

Cross Lingual Embeddings

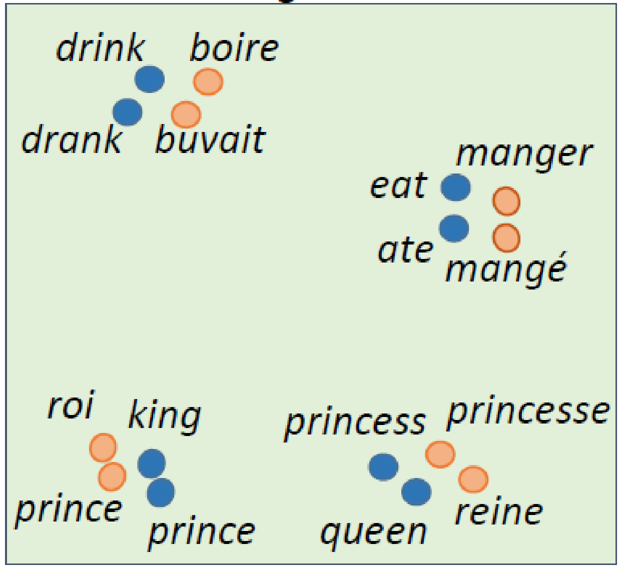
English



French



Joint English French



Monolingual Word Representations
(capture syntactic and semantic similarities between words)

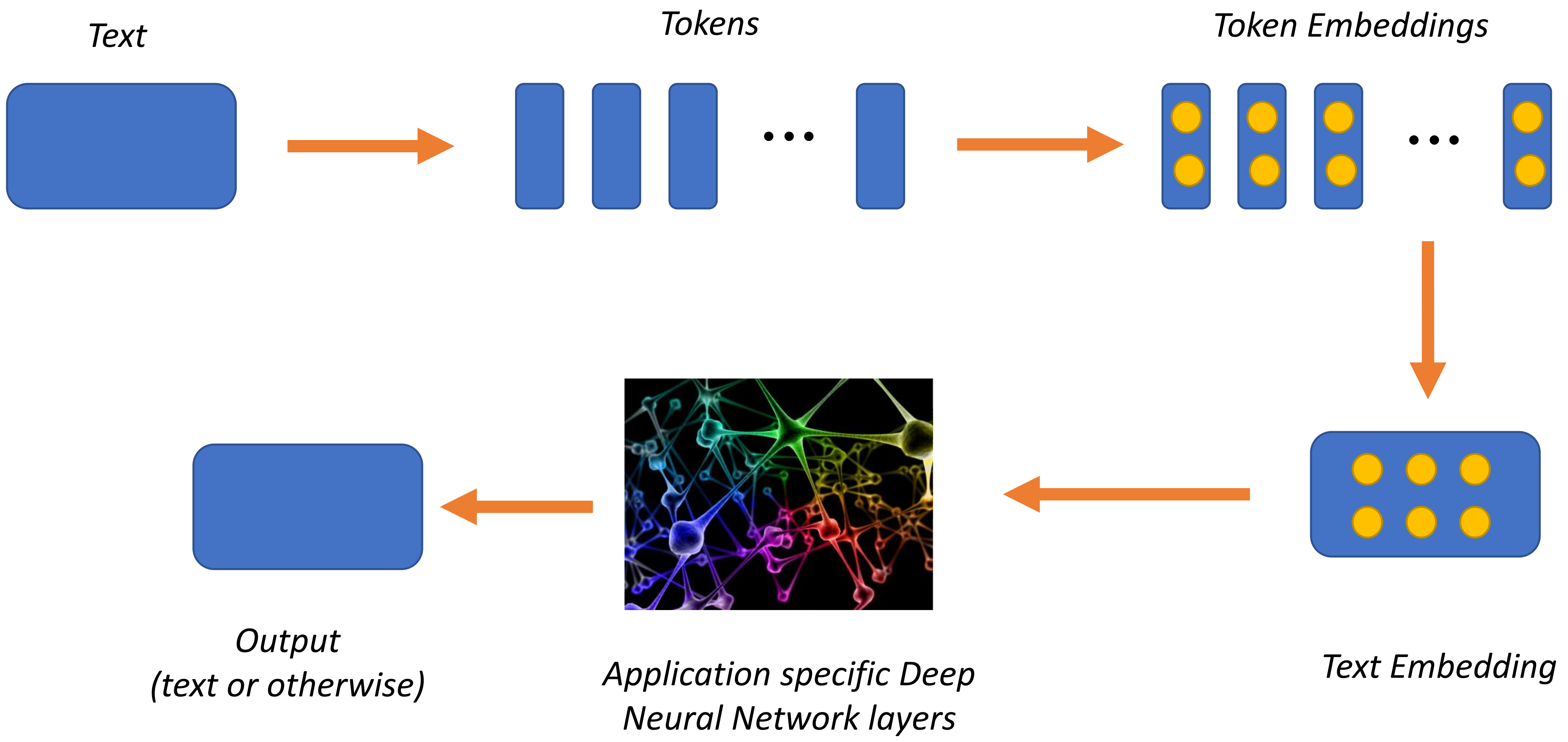
Multilingual Word Representations
(capture syntactic and semantic similarities between words both within and across languages)

(Source: Khapra and Chandar, 2016)

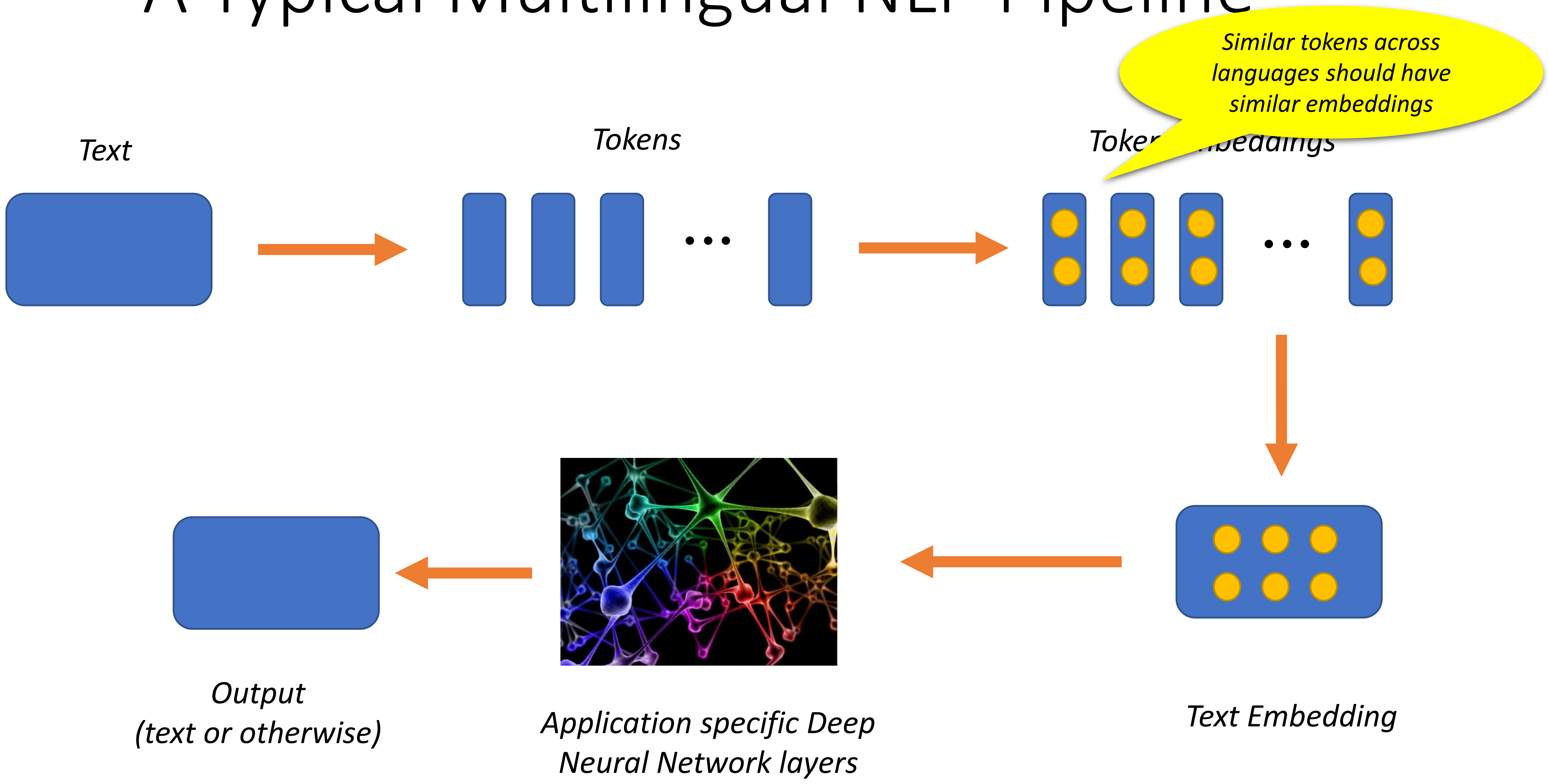
$$embed(y) = f(embed(x))$$

x, y are source and target words
$embed(w)$: embedding for word w

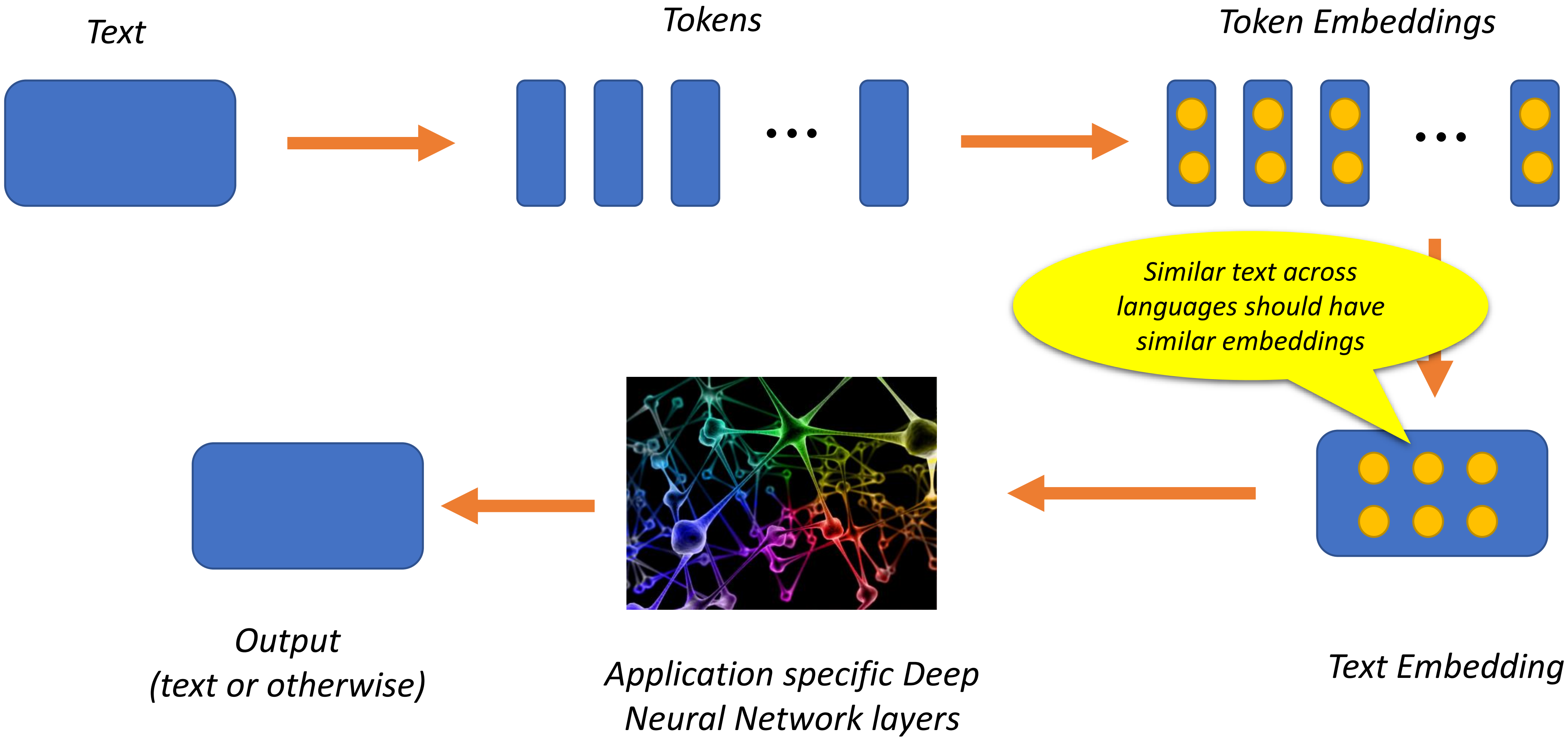
A Typical Multilingual NLP Pipeline



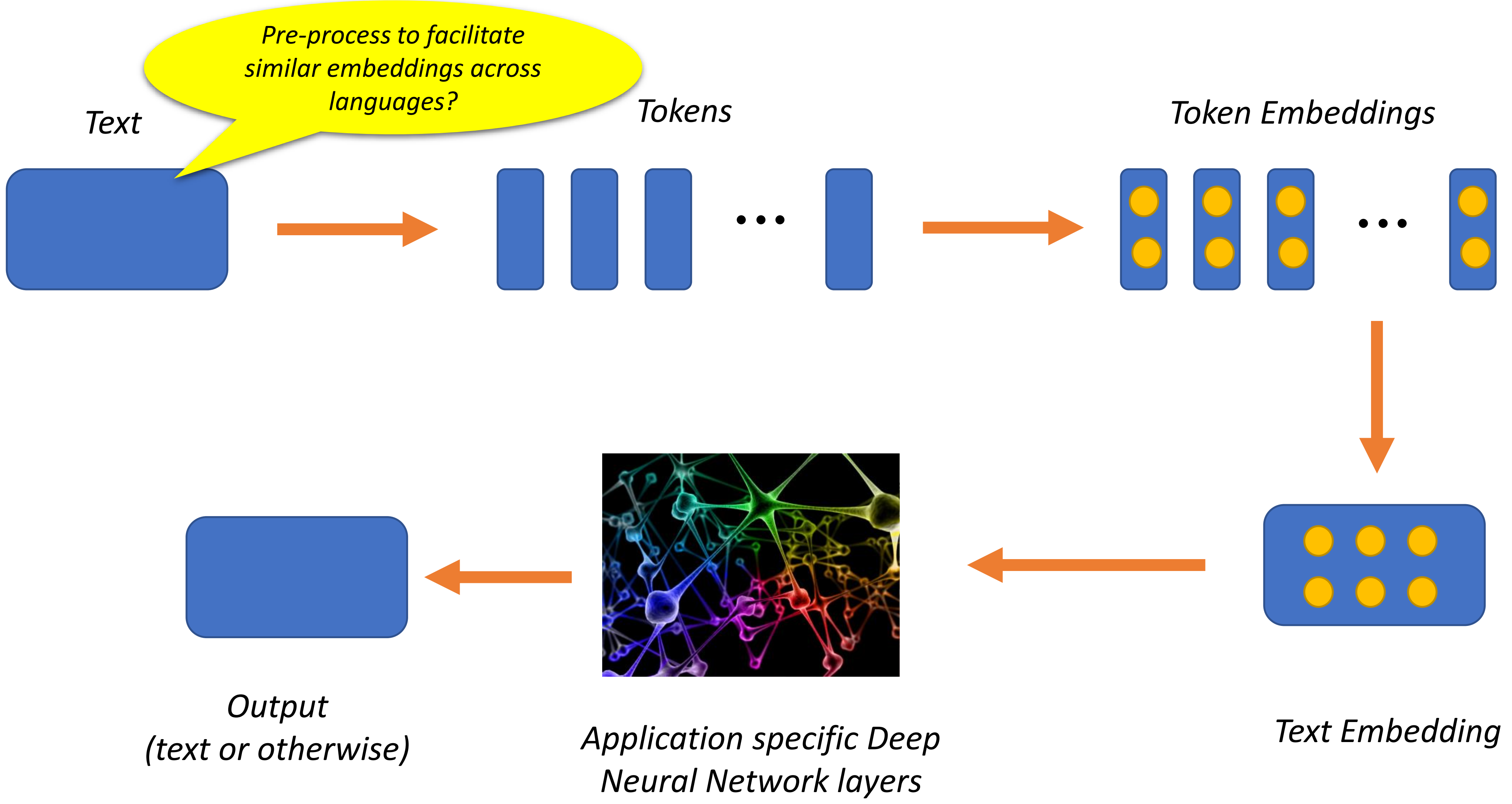
A Typical Multilingual NLP Pipeline



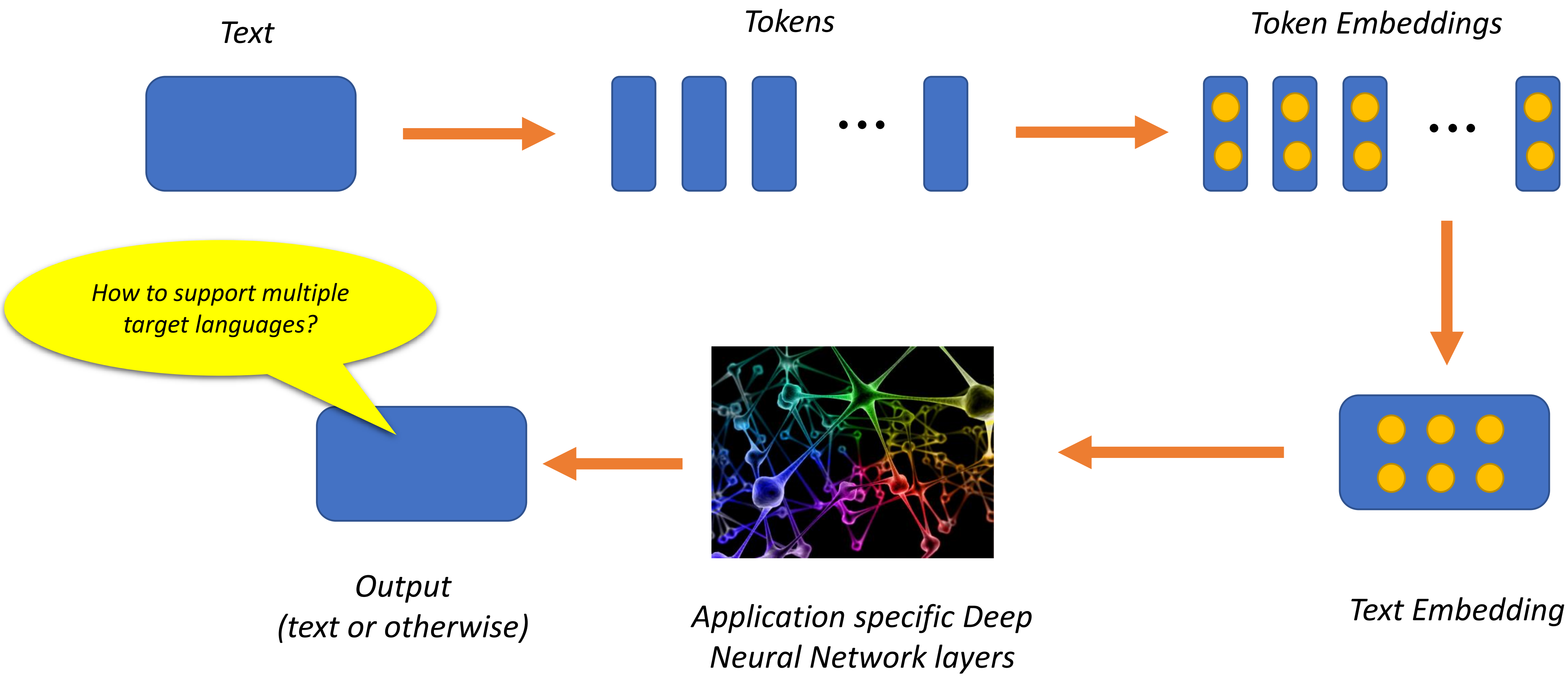
A Typical Multilingual NLP Pipeline



A Typical Multilingual NLP Pipeline



A Typical Multilingual NLP Pipeline



More Reading Material

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

SMT Tutorials

- *Machine Learning for Machine Translation (An Introduction to Statistical Machine Translation)*. **Tutorial at ICON 2013** with Prof. Pushpak Bhattacharyya, Piyush Dungarwal and Shubham Gautam. [slides] [handouts]
- *Machine Translation: Basics and Phrase-based SMT*. **Talk at the Ninth IIIT-H Advanced Summer School on NLP (IASNLP 2018), IIIT Hyderabad** . [pdf] [pptx]

NMT Tutorial

Machine Translation for Related Languages

- *Statistical Machine Translation between related languages. Tutorial at NAACL 2016 with Prof. Pushpak Bhattacharyya and Mitesh Khapra.* [abstract] [slides]
- *Machine Translation for related languages. Tech Talk at AXLE 2018 (Microsoft Academic Accelerator). [pdf] [pptx]*
- *Translation and Transliteration between related languages. Tutorial at ICON 2015 with Mitesh Khapra. [abstract] [slides] [handouts]*

Multilingual Training

- *Multilingual Learning. Invited Talk at IIIT Hyderabad Machine Learning Summer School (Advances in Modern AI) 2018. [slides]*

Thank you!

anoop.kunchukuttan@gmail.com