Natural Language Processing and Machine Translation

Machine Translation

Abhishek Koirala

M.Sc. in Informatics and
Intelligent Systems
Engineering

THAPATHALI CAMPUS
INSTITUTES OF ENGINEERING

Introduction to machine translation

- A platform that uses various forms of artificial intelligence to automatically translate context
- Basic idea is that no human intervention should be needed during translation
- Machine Translation might not be very accurate in the beginning, but it can start translation immediately and automatically

Types of Machine Translation

Four most common machine translations

- Rule based machine translation
- Corpus based machine translation
- Statistical machine translation
- Neural Machine Translation

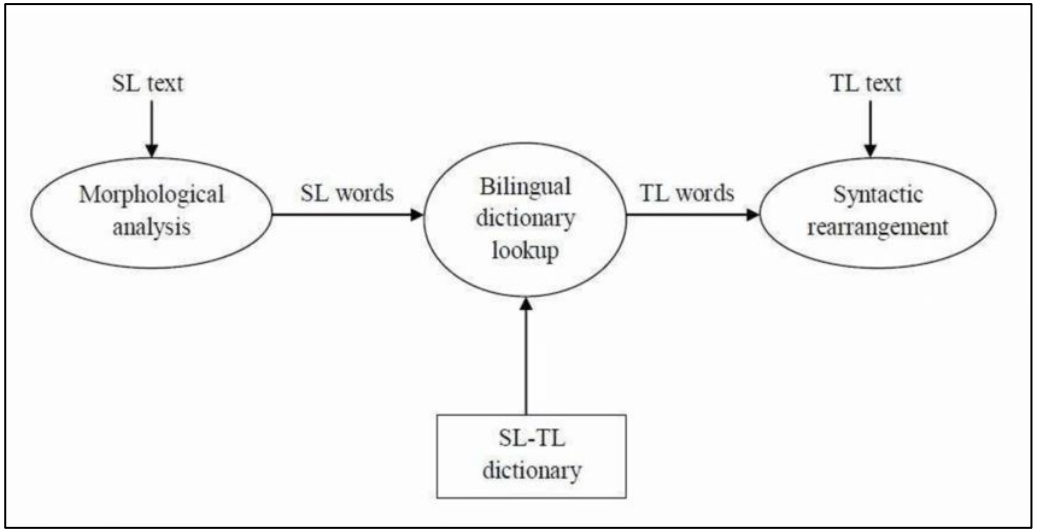
Rule based Machine Translation

- Relies on context such as linguistic and grammatical rules
- Was first commercial machine translation systems that allows words to be put in different places to have different meanings depending on context
- Rules are developed by human language experts and programmers (Limitation ?)
- Relies on manually built translation lexicons, which can be edited and refined by users to improve the translation
- Any refinements in rules are time consuming to implement and maintain and in some cases can lead to lower quality translations due to ambiguity of terms

Rule based Machine Translation

Direct Machine Translation

- Most elementary form of MT
- Using simple rule structure, direct MT breaks the source sentences into words, compares them to the inputted dictionary, then adjusts the output based or morphology and syntax
- Time intensive method
- Difficult to incorporate all words in lexicon
- Great start to MT, but quickly replaced by more advanced techniques



Transfer Based Machine Translation

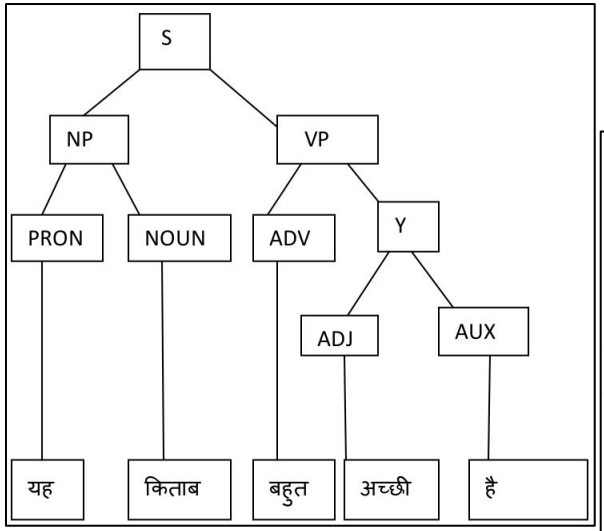
- It forgoes a word by word translation , first analyzing a source language's grammar structure
- Broken down into 3 steps
  - **Analysis**: machine analyzes source language to identify grammatical rule set
  - **Transfer**: Sentence structure is then converted into a form that is compatible with the target language
  - **Generation**: Once a suitable structure has been determined, the machine produces a translated text
- The approach still used a word substitution format, limiting its scope of use
- While streamlining grammatical rules, it also increased the number of word formulas compared to direct machine translation.

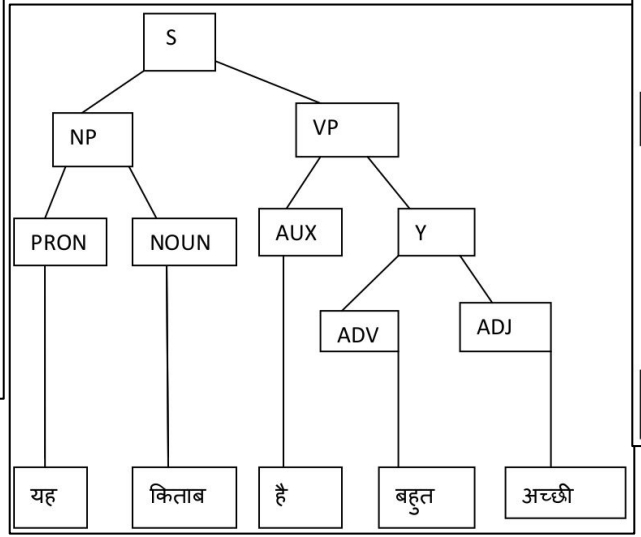
Transfer based Machine Translation

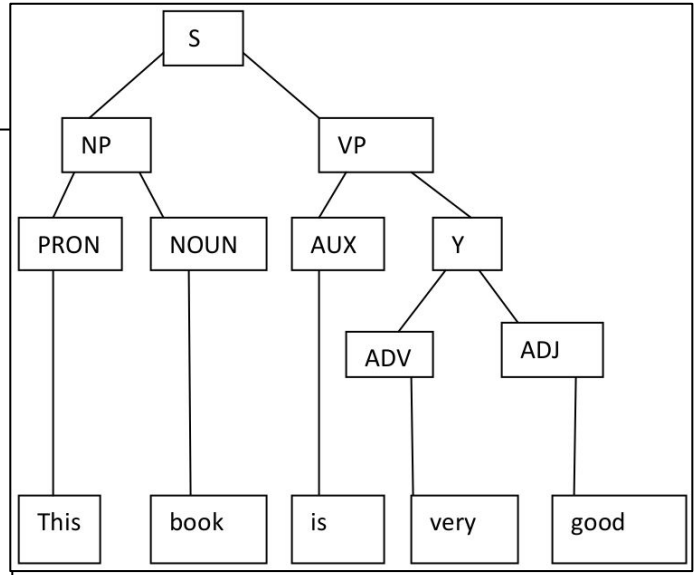
Hindi Structure	English Structure
S ->NP VP	S->NP VP
NP->PRON NOUN	NP->PRON NOUN
VP->ADV Y	VP->AUX Y
Y->ADJ AUX	Y->ADV ADJ

Input:यह किताब बहुत अच्छी है

Output:This book is very good





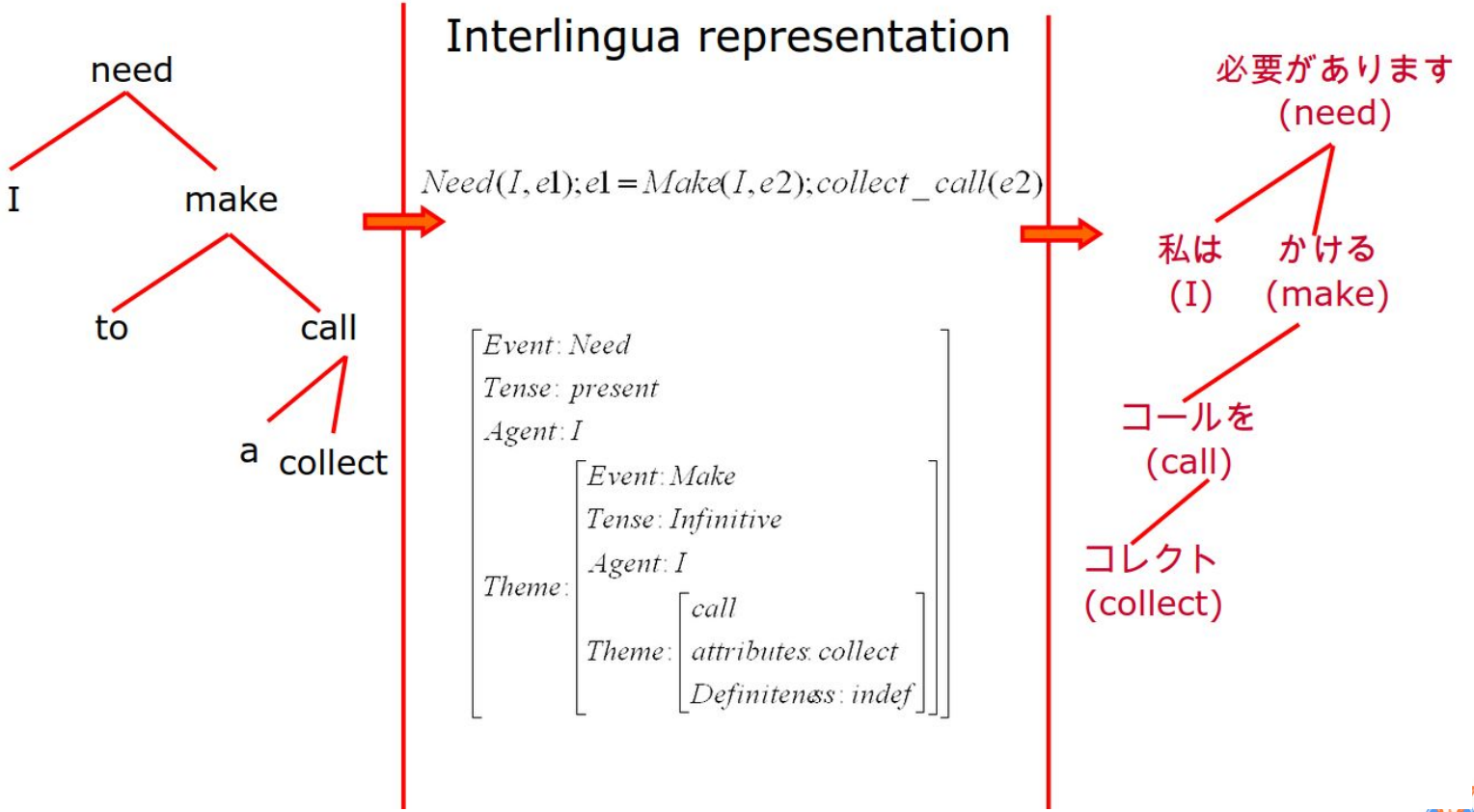


Hindi to English Transfer based Translation System : https://arxiv.org/pdf/1507.02012.pdf

Interlingual Machine Translation

- Process of translating text from source language to interlingua (an intermediate artificial language developed to translate words and meanings from one language to another)
- This method is sometimes mistaken for a transfer based MT system. However, interlingual MT provides a wider range of application. Because the source language is converted to interlingua, it can include multiple target languages.
- Major benefit of this approach is that developers only need to create rules between a source language and interlingua
- Drawback is that creating an all-encompassing interlingua is extremely challenging

Interlingual Machine Translation



Rule Based Machine Translation

Pros and Cons

- Main benefit of RBMT is that translation can be reproduced. The rules dictating the translations account for morphology, syntax and semantics, even if the translation isn't clear, it will always come back the same.
- Allows linguists and programmers to tailor the application for specific use cases.
- Main drawback is that every language includes subtle expressions and dialects. Countless rules and thousands of language pairs need to be factored into the application.
- Rules need to be constructed around a vast lexicon, considering each word's independent morphological, syntactic and semantic attributes.

Corpus based Machine Translation

- Huge collection of data and texts of both source and target language is needed.
- The parallel collected corpus is employed for the purpose of translation
- **Example based MT**
  - Mapping between source language and target language
  - Memory based translations
  - Stores the input and matches the sentence to be translated with what is already available in the system
  - If they match well, correct translation will be generated
  - Depends on concept of analogy

Example based Machine Translation

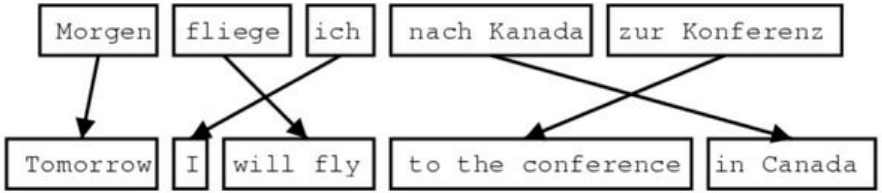
- Uses side-by-side, phrase-to-phrase, parallel texts(bilingual corpus) as its core framework
- Idea is phrase-to-phrase method would produce a better translation. The more phrases you add to the database, the easier it is for the system to find a substitute word
- If the phrase “I want to eat something” has already been translated into target language, then translating “I want to drink something” doesn’t require full sentence to be translated word-for-word.
- We only need to decipher the differences between the phrases, look up the unknown words, and hope and exception does not exist.
- Enhanced the accessibility of MT, because complex language rules are generally already built into each phrase.

Statistical Machine Translation

- IBM's Thomas J. Watson Research Center showcased a MT system
- Does not rely on rules or linguistics for its translations. Instead the system approaches language translation through analysis if patterns and probability
- Comes from a language model that calculates the probability of a phrase being used by a native language speaker. It then matches two languages, that have been split into words, comparing the probability that a specific meaning was intended
- Translations are based on the context of the sentence rather than rules.
- More accurate and less costly than RBMT and EBMT systems
- When SMT is first created, all translations are given equal weight. As more data is entered into the machine to build patterns and probabilities, the potential translation begins to shift.

Statistical Machine Translation

- How to learn $P(x|y)$?
- First, need large amount of parallel data (i.e., French and English)
- Define alignment a, i.e., a word-level correspondence between source sentence target sentence y



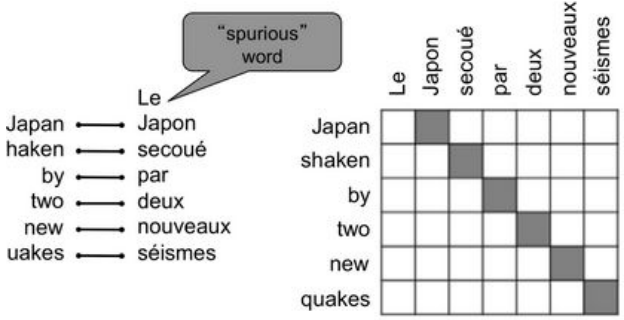
- Redefine the probability which includes the alignment

$$P(x, a|y)$$

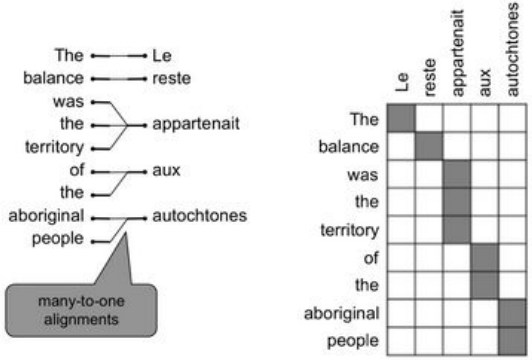
Alignment

- **Alignment** is the **correspondence between particular words** in the translated sentence pair
  - Note: some words have no counterpart, some have many-to-one relationships,

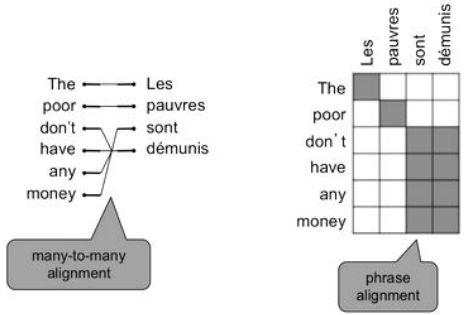
No counterparts



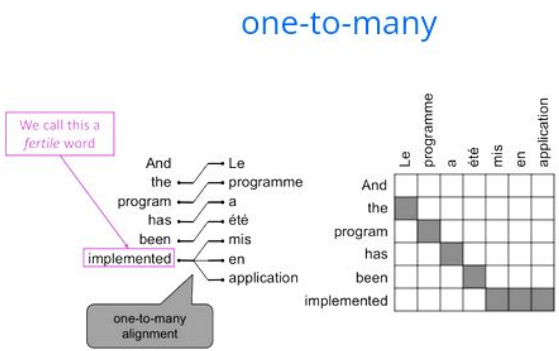
many-to-one



many-to-many



one-to-many



Examples from: "The Mathematics of Statistical Machine Translation: Parameter Estimation", Brown et al, 1993. http://www.aclweb.org/anthology/J93-2003

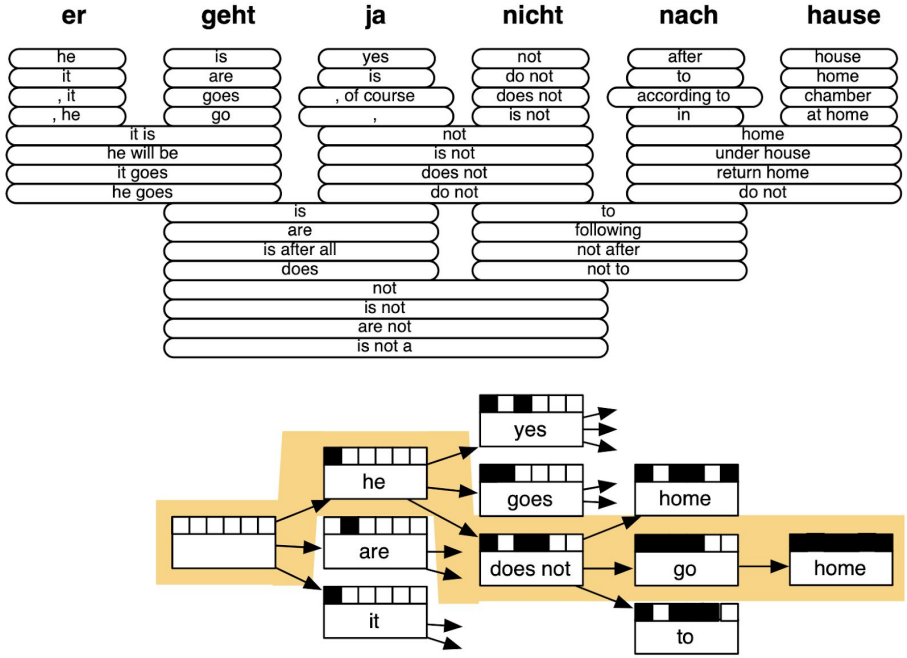
THAPATHALI CAMPUS
INSTITUTES OF ENGINEERING

Learning Alignment

- We learn $P(x, a|y)$ as a combination of many factors:
  - Probability of particular words aligning
  - Probability of particular words on a position
  - Probability of particular words having a particular fertility
  - Etc.
- Alignments a are discrete, not probabilistic
  - Require the use of special learning algorithms (like Expectation-Maximization such as in k-means algorithm) for learning the parameters of distributions

Decoding for SMT

- We could enumerate every possible y and calculate the probability -> Too expensive!
- Answer: Use some dynamic programming (e.g., **viterbi**) to figure out **optimal paths (this is actually very slow!)**



Statistical Machine Translation, Chapter 6, Koehn, 2009.
https://www.cambridge.org/core/books/statistical-machine-translation/94EADF9F680558E13BE759997553CDE5

1990s-2010s: SMT

- SMT was a **huge research field**
- The best systems were extremely complex
  - Hundreds of important details we haven't mentioned here
  - Systems had many **separately-designed subcomponents**
  - **Lots of feature engineering**
    - Need to design features to capture particular language phenomena
  - Require compiling and maintaining **extra resources**
    - Like tables of equivalent phrases
  - Lots of **human effort** to maintain
    - Repeated effort for each language pair

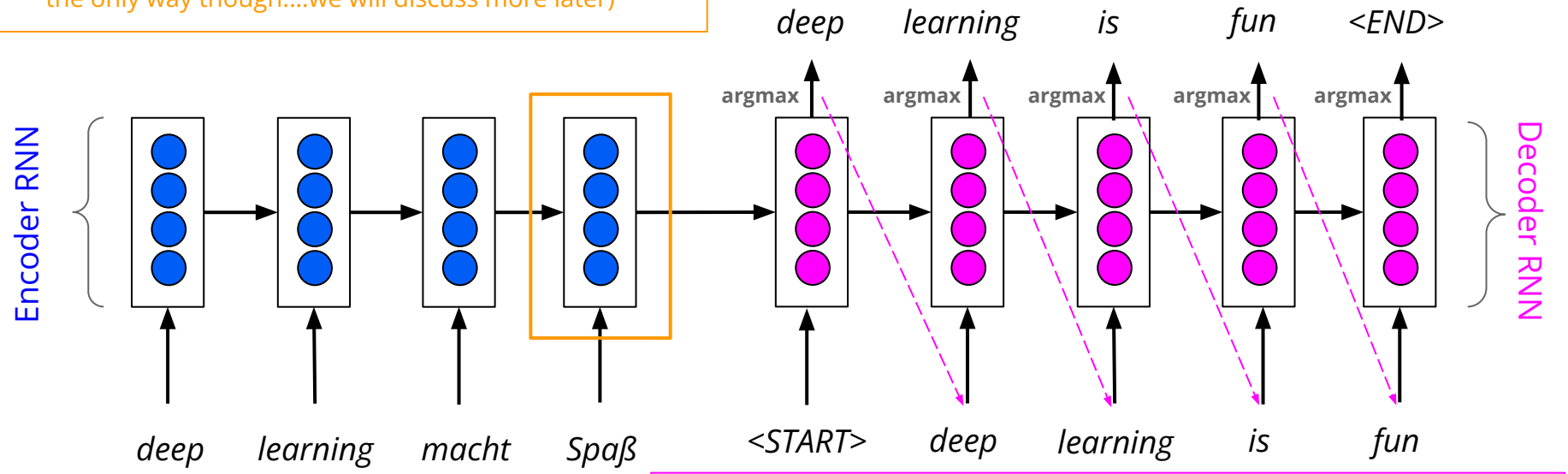
Neural Machine Translation

- **Neural Machine Translation** (NMT) is a way to do Machine Translation with a **single end-to-end neural network**
- The neural network architecture is called a **sequence-to-sequence model** (aka **seq2seq**) and it involves **two** RNNs

Sequence to sequence model

Encoding of the whole source sentence. Input the final hidden state for Decoder RNN. (Using the final hidden state is not the only way though....we will discuss more later)

Note that this diagram is showing the generation (testing) mode. The training mode will be in "teacher forcing" mode instead.

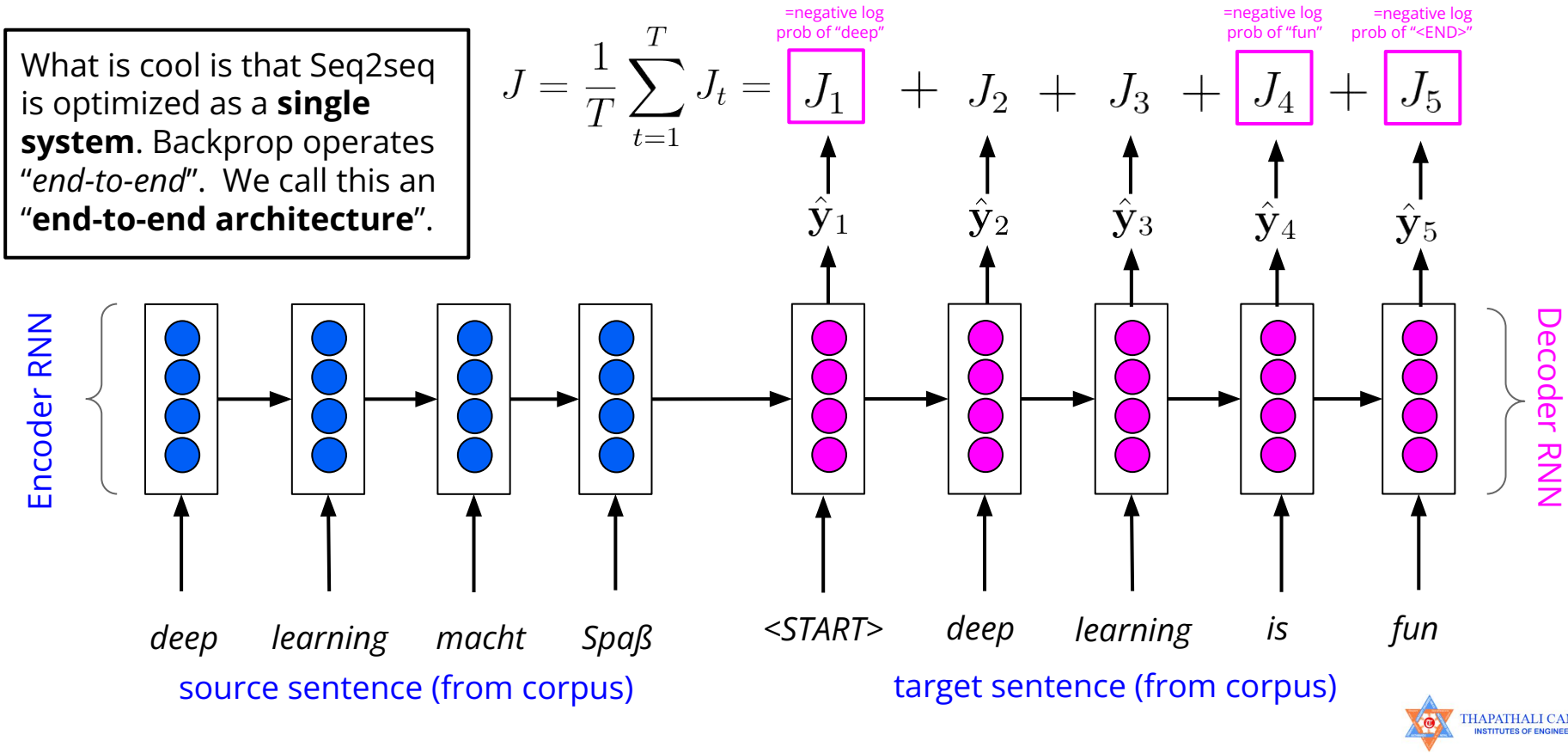


Encoder RNN produces an encoding of the source sentence.

Decoder RNN is a LM that generates target sentence, conditioned on ending. It does NOT need to be same length as the encoder. It takes in the <START> token, and will stop when it meets the <END> token or the specified max length.

Training NMT System

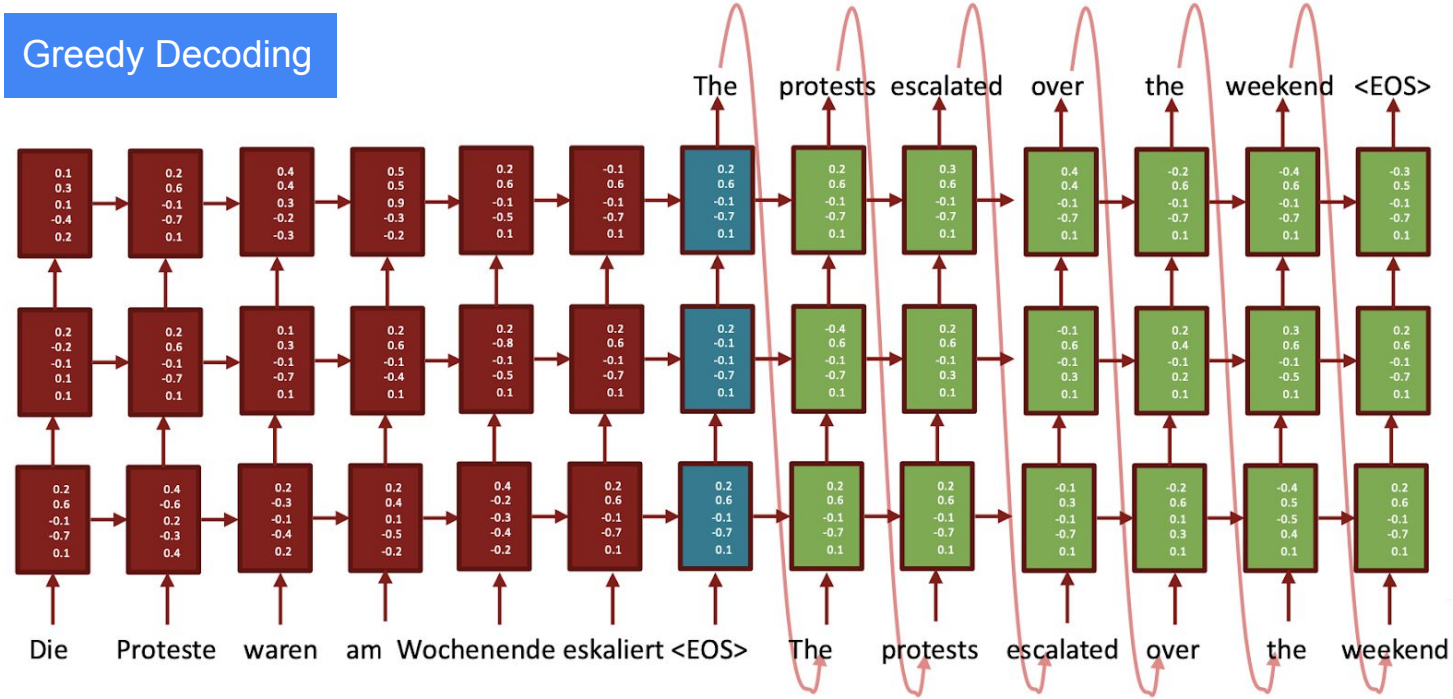
What is cool is that Seq2seq is optimized as a **single system**. Backprop operates "*end-to-end*". We call this an "**end-to-end architecture**".



THAPATHALI CAMPUS
INSTITUTES OF ENGINEERING

Multilayer (Stacked) NMT

Greedy Decoding



Sequence to Sequence Learning with Neural Networks, Sutskever et al. 2014, https://arxiv.org/abs/1409.3215

Beam Search

- **Beam search**: on each step of decoding, keep track of the *k* most probable partial translations (which we call **hypothesis**)
  - *k* is the beam size (in practice around 5 to 10)

- A hypothesis $y_1, \cdots, y_t$ has a **score** which is its log prob

$$\text{score}(y_1, \cdots, y_t) = \log P_{\text{LM}}(y_1, \cdots, y_t | x) = \sum_{t=1}^{t} \log P_{\text{LM}}(y_i | y_1, \cdots, y_{i-1}, x)$$

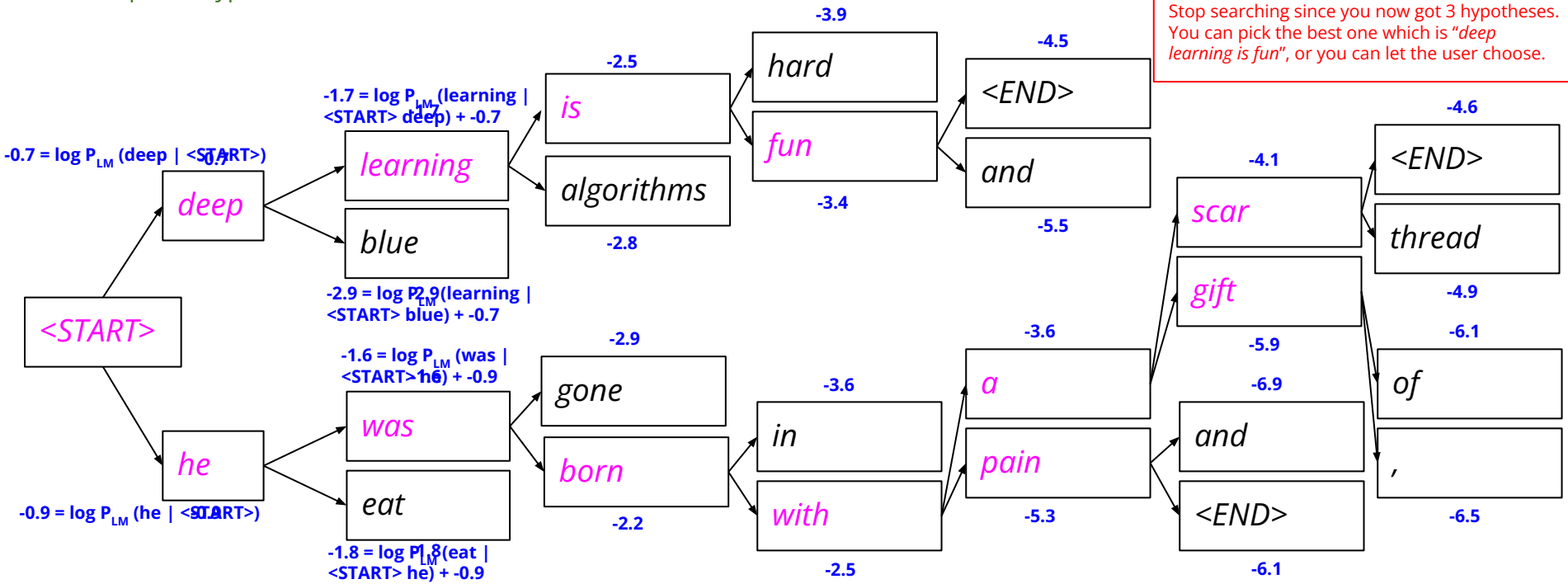
  - Scores are all **negative**, and higher score is better
  - We search for high-scoring hypotheses, tracking top *k* performers

- Beam search is **not guaranteed** to find optimal solution
- But **much more efficient** than exhaustive search!

Beam Search - Example

Beam size = k = 2 (numbers of parallel search). Blue numbers = $\text{score}(y_1, \cdots, y_t) = \log P_{\text{LM}}(y_1, \cdots, y_t|x)$

no.of required hypo = n = 3. Pink refers to the search words



Beam Search - stopping criterion

- In **greedy decoding**, usually we decode until the model produces an <END> token For example: <START>deep learning is fun<END>

- In **beam search decoding**, different hypotheses may produce <END> tokens on different timesteps
  a. When a hypothesis produces <END>,that hypothesis is complete.
  b. Place it aside and continue exploring other hypotheses via beam search

- Usually we continue beam search **until**:
  a. We reach time step T (where T is some predefined cutoff),or
  b. we have at least n completed hypotheses (where n is pre-defined cutoff)

- Potential problem: longer sentences tend to have lower scores
  i. Fix: normalize by length (divide by length)

Neural Machine Translation

NMT Pros:

- Better performance
  a. More **fluent**
  b. Better use of **context**
  c. Better use of **phrase similarities**
- A single network that does all
  a. Less human engineering effort

NMT Cons:

- Less **interpretable**
  a. Hard to debug
- Difficult to **control**
  a. Difficult to impose policy or rules

NMT: Biggest success in NLP?

NMT went from **fringe research attempt** in 2014 to the **leading standard method** in 2016

- **2014**: First seq2seq paper published
- **2016**: Google Translate switches from SMT to NMT and by **2018** everyone switches (Microsoft, Tencent, Baidu, Facebook, etc.)

It is kind of sad that SMT systems were built by **hundreds** of engineers over **many** years, but were outperformed by NMT trained by **small** number of engineers in a **few** months :(

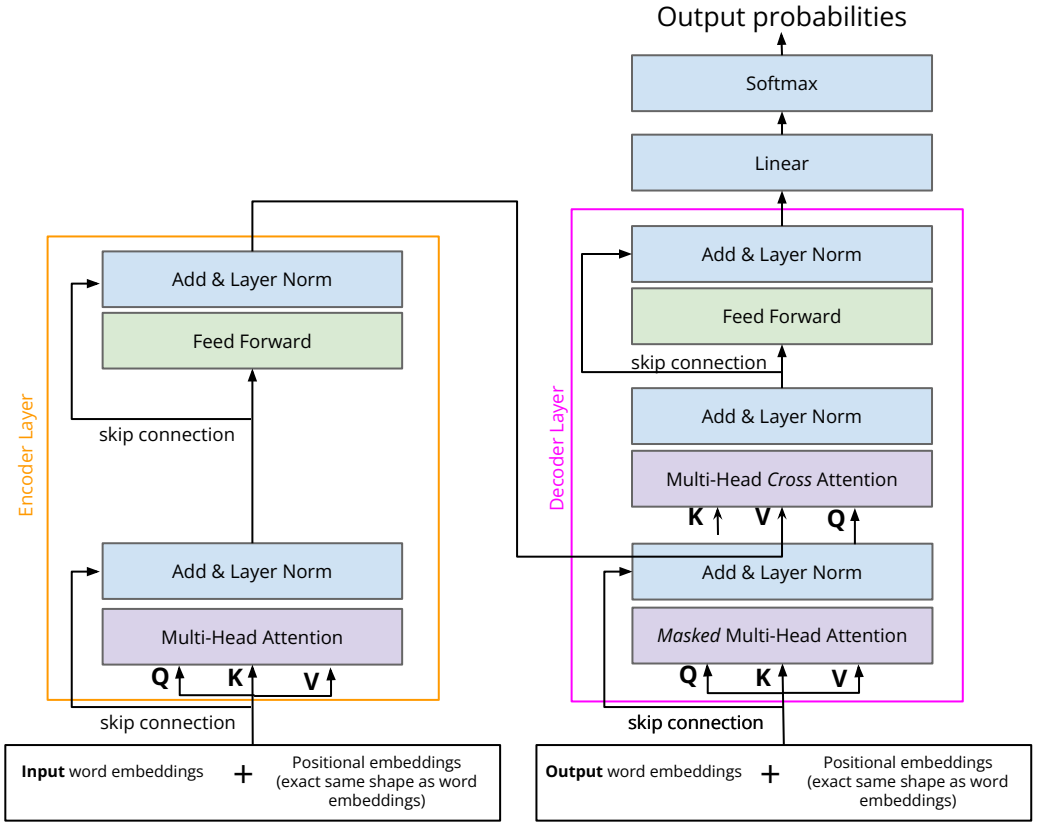
THAPATHALI CAMPUS
INSTITUTES OF ENGINEERING

Is NMT solved?

Many difficulties remain:

- **Out-of-vocabulary**
- **Domain mismatch** between train and test data
- Maintaining **context** over longer text
- **Lower-resource** language pairs
- Failures to accurately capture **sentence meaning**
- **Pronoun** (or zero pronoun) **resolution** errors
- **Morphological agreement** errors

Transformers [Vaswani et al., NeurIPS 2017]



- Originally developed for **seq2seq** tasks, e.g., MT
- **Idea**: Use **attention ONLY** (without RNN) will
  - enable **parallelizability** hence more efficient
  - learn **long-range dependencies** better
  - provides **interpretability**
- **Obstacles**:
  - **Loss of sequential** information
  - The **decoding side** could be a problem if we do everything at the same time, because the input will see the answer (unlike the sequential RNN where each set of input is used to predict the output)
  - Need to be careful that the whole architecture is **always parallelizable at almost every step**

Attention Is All You Need, Vaswani et al. 2017, https://arxiv.org/pdf/1706.03762.pdf

THAPATHALI CAMPUS
INSTITUTES OF ENGINEERING


thank you
danke
謝謝
ngiyabonga
شكراً جزيلاً
рахмат
teşekkür ederim
Баярлалаа
спасибо
gracias
merci
obrigado
dziękuję
bedankt
sukriya
kop khun krap
go raibh maith agat
mochchakkeram
tapadh leat
arigatô
dakujem
terima kasih
grazie
감사합니다
ありがとう
мерси
хвала
ευχαριστώ
mersi
barka
welalin
tack
spas
dank je
misaotra
matondo
paldies
grazzi
mahalo
vinaka
спасибі
blagodaram
faafetai lava
kiitos
dankie
dhanyavad
nanni
nandri
mauruuru
köszönöm
hvala
gràcie
bayarlalaa
enkosi
asante
manana
obrigada
tenki
murakoze
djiere dieuf
tau
дякую
mamnun
chokrane
chnorakaloutioun
gratias ago
gràcies
sulpáy
taiku
sobodi
dēkuji
sagolun
mèsi
didi madloba
kam sah hamnida
najis tuke
rahmat
rahmet
tanemirt
diolch
dhanyavadagalu
shukriya
merce
trugarez
takk
xièxie
akun
dankon
aciū