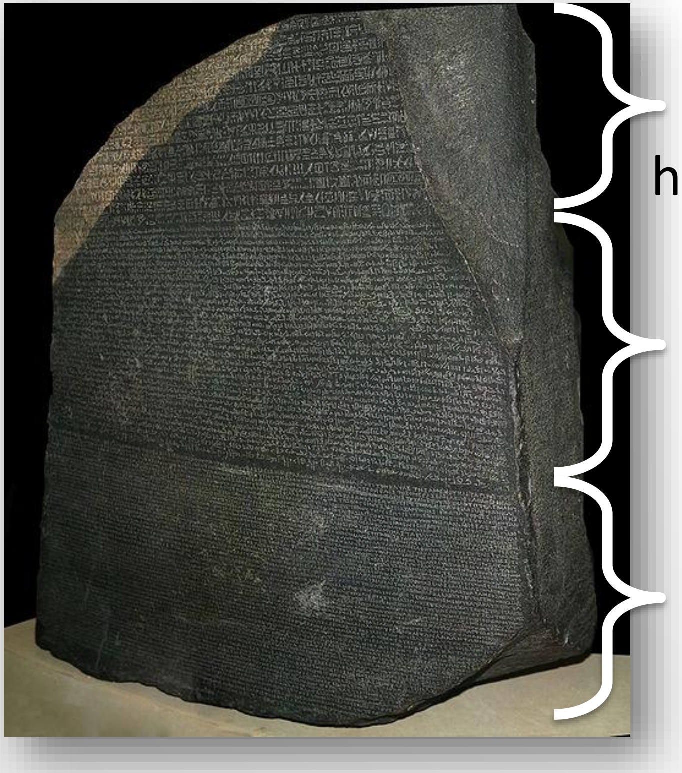
Image: Trawling for Babel fish. Concept and juxtaposition: Raeid Saqur.

Statistical + **Neural machine translation**

CSC401/2511 – Natural Language Computing – Spring 2026
Ken Shi and Gerald Penn
Lecture 6
University of Toronto

The Rosetta Stone

- The **Rosetta Stone** dates from 196 BCE.
  - It was re-discovered by French soldiers during Napoleon’s invasion of Egypt in 1799 CE.



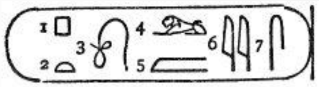
Ancient Egyptian hieroglyphs

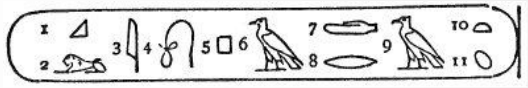
Egyptian Demotic

Ancient Greek

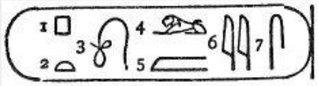
- It contains three **parallel** texts in different languages.
- Demotic had been partly deciphered.
- *For 20+ years after Rosetta’s discovery, Egyptian hieroglyphics largely remained a mystery.*

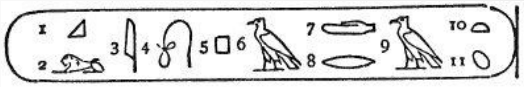
Deciphering Rosetta

- During 1822–1832, **Jean-François Champollion** worked on the Rosetta stone. He noticed:
  1. The circled Egyptian symbols, e.g.  appeared in roughly the same positions as words like ‘*Ptolemy*’ in Greek.

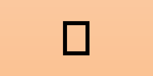
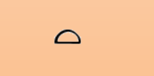
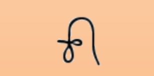
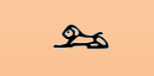
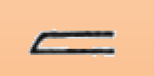
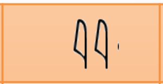
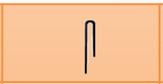
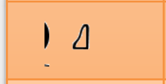
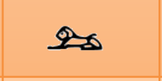
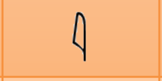
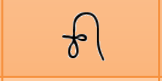
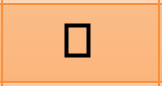
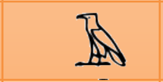
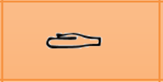
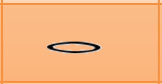
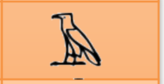
  2. The number of Egyptian hieroglyph tokens was **much larger** than the number of Greek words → Egyptian seemed to have been partially phonographic.
  3. Cleopatra’s cartouche was written 

Deciphering Rosetta

- So if 

was ‘*Ptolemy*’ and 

was ‘*Cleopatra*’ and the symbols corresponded to sounds – can we match up the symbols?

								
P	T	O	L	M	E	S		
								
C	L	E	O	P	A	T	R	A

- This approach demonstrated the value of working from **parallel texts** to decipher an unknown language:
  - *There are several examples of decipherment having been achieved without aligning unknown words in bitexts.*

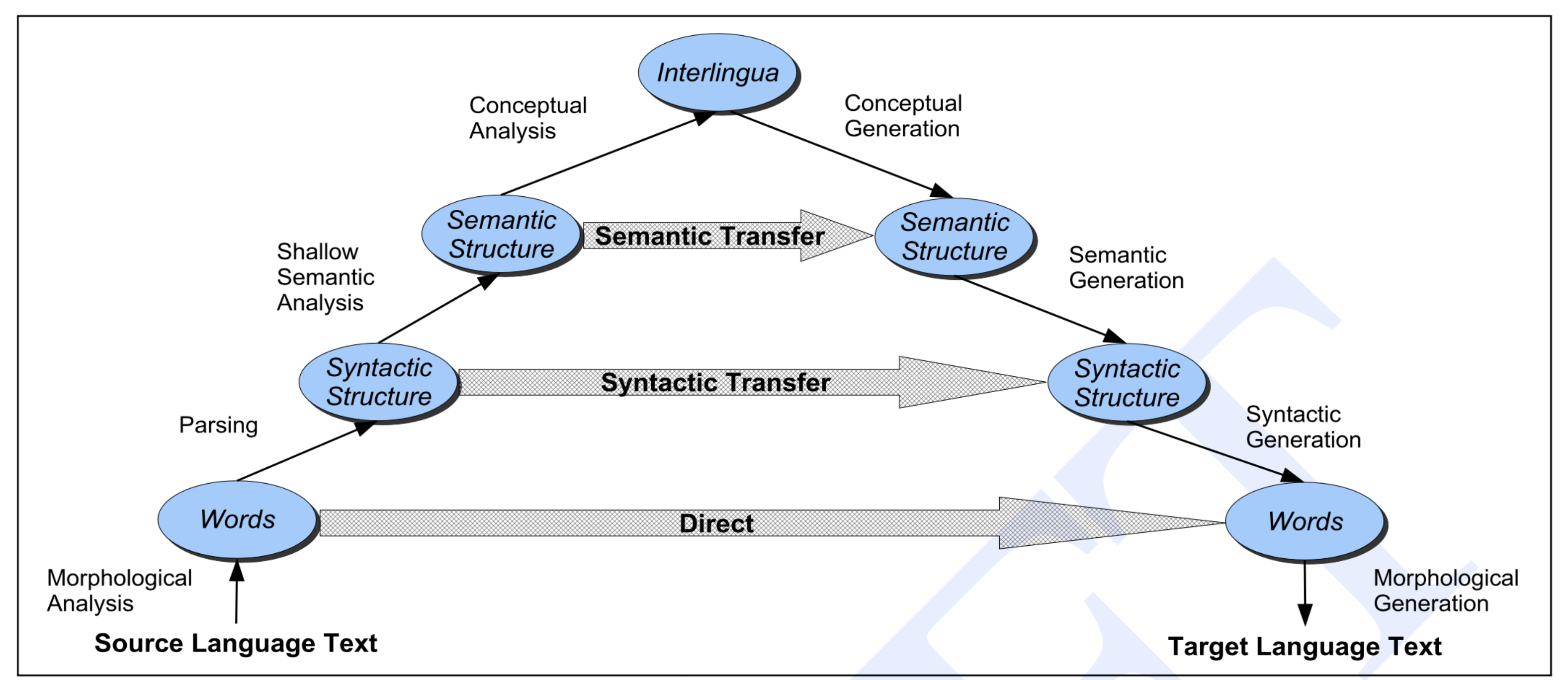
Circa 2016

- What happened to my machine translation (SMT)?
  - ABC’s speech recognizer transcribes French as though it were English in the Prime Minister’s bilingual remarks:

“…Nazi innings…
…recourse to ice packs
…I’d love the log trucks”

The Vauquois triangle (1968)

- High-level classes of methodologies:
  - “Direct” Translation
  - Syntactic Transfer
  - Semantic Transfer
  - Interlingua

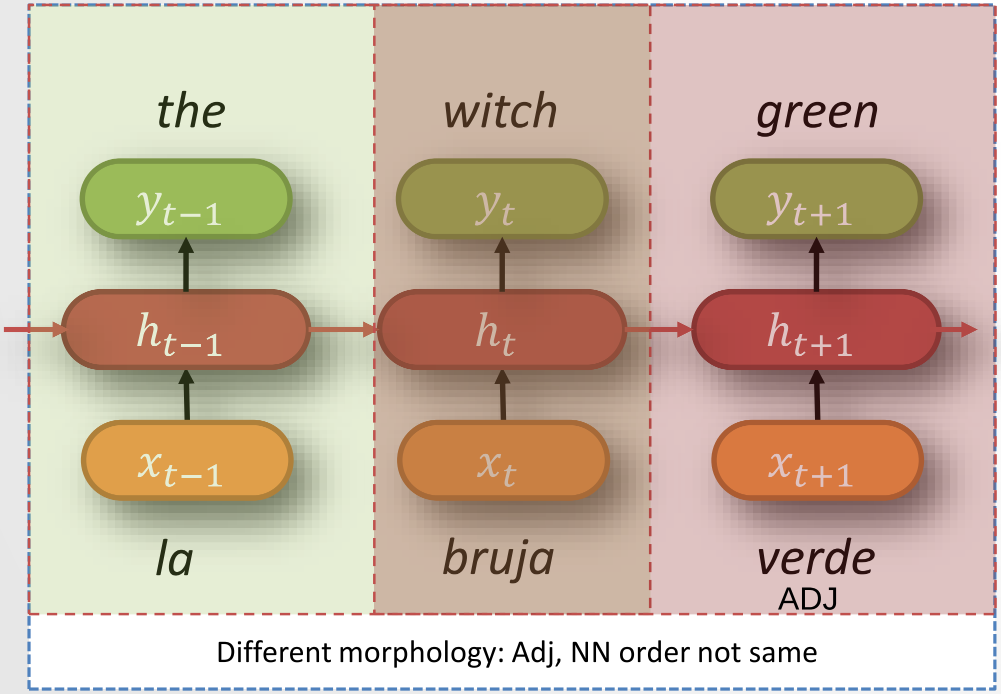


Seq2seq motivation

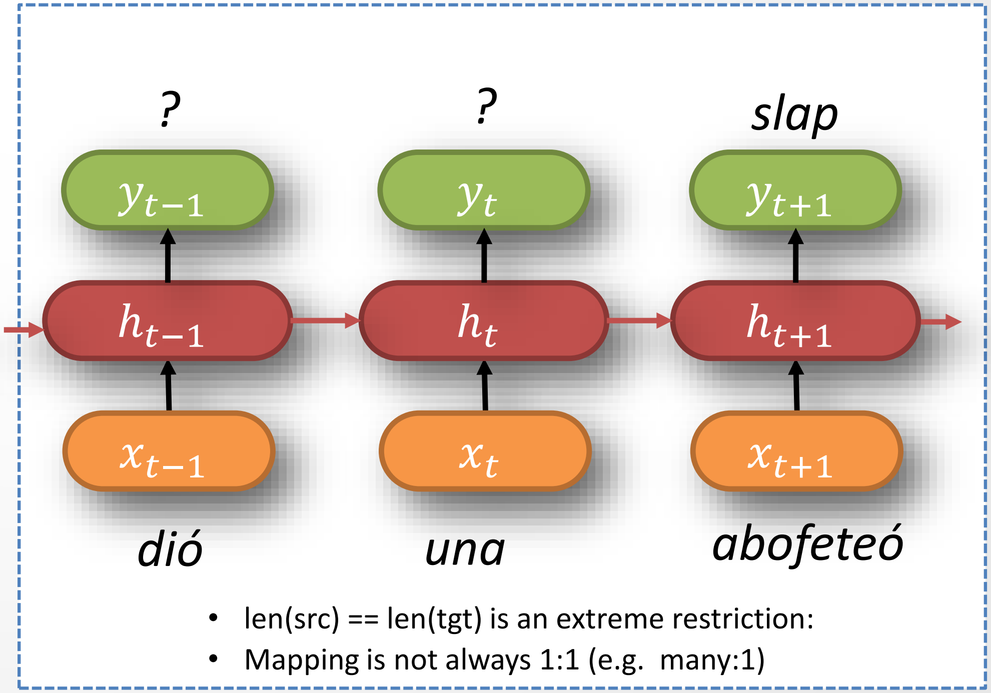
In the absence of any hierarchical structure (such as from a parser), there aren’t a lot of options that are invariant to the choice of language pair.

Why not train an RNN to output a translated token from source token?

“*Mary no **dió una abofeteó** a la bruja verde.*” -> “*Mary did not **slap** the green witch.*”



Different morphology: Adj, NN order not same



- len(src) == len(tgt) is an extreme restriction:
- Mapping is not always 1:1 (e.g. many:1)

“Direct” translation

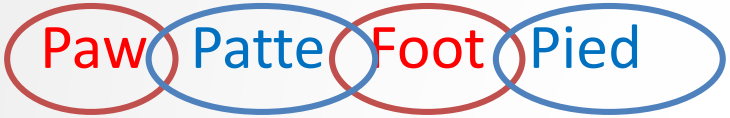
- A bilingual dictionary that aligns words across languages can be helpful, but only for certain cases.

¿	*Dónde*	*está*	*la*	*biblioteca*	*?*
	Where	*is*	*the*	*library*	*?*
	Où	*est*	*la*	*bibliothèque*	*?*

Mi	*nombre*	*es*	*T-bone*
My	*name*	*is*	*T-bone*
Mon	*nom*	*est*	*T-bone*

Difficulties in MT: ambiguity

- **Ambiguity** makes it hard to pick one translation
  - Lexical: many-to-many word mappings



Paw Patte Foot Pied

  - Syntactic: same token sequence, different structure
    - Rick hit the Morty [with the stick]$_{PP}$ / Rick golpeó el Morty con el palo
    - Rick hit the Morty [with the stick]$_{PP}$ / Rick golpeó el Morty que tenia el palo
  - Semantic: same structure, different meanings
    - I’ll pick you up / {Je vais te chercher, Je vais te ramasser}
  - Pragmatic: different contexts, different interpretations
    - Poetry vs technical report

Difficulties in MT: typology

- Different **morphology** → difficult **mappings**, *e.g.*
  - Many (*polysynthetic*) vs one (*isolating*) roots per word

    e.g., Yupik e.g., Cantonese
  - Many (*fusional*) vs few (*agglutinative*) *features* per morpheme

    e.g., Russian e.g., Turkish
- Different **head-position effects in syntax**, *e.g.*
  - SVO vs. SOV vs. VSO (e.g. English vs. Japanese vs. Arabic)
    - He listens to music / kare ha ongaku wo kiku

      Subject Verb Object Subject Object Verb
  - Satellite vs. nuclear-framed (e.g. Spanish vs. English)
    - La botella salió flotando / The bottle floated out

“Statistical Machine Translation”

- 1989-2014: SMT was a huge research field. All pre-neural.
- Best systems were extremely complex with many separately designed sub-components
- Lots of human effort & optimization for specific language pairs (e.g. SBMT for Arabic-English, PBMT for Chinese-English)
- Rule-based, hand-designed components never really were replaced in their entirety (e.g., headedness of NPs)

NMT – the breakout of Deep Learning in NLP

- Although there had been significant advances in neural language modelling and neural acoustic modelling beforehand, the NLP community remained resistant to embracing neural methods until a wildly successful attempt at neural MT in 2014. [1,2]
- NMT systems trained by a small group of engineers in a few months outperformed a state-of-the-art heavily engineered SMT system.
- Even now, NMT remains an important rationalizer for neural methods in NLP – it was one of the first showcase tasks for attention mechanisms.

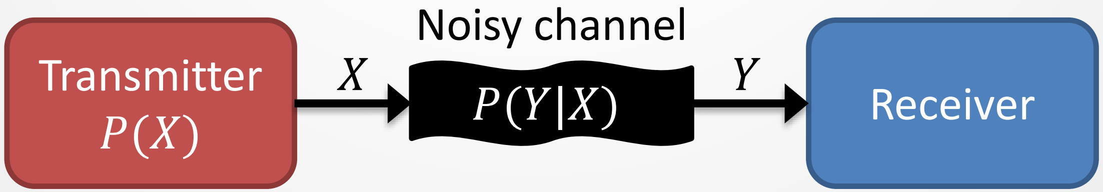
[1] Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R., & Makhoul, J. (2014). Fast and Robust Neural Network Joint Models for Statistical Machine Translation. *Proc. ACL,* pp. 1370–1380.

The "noisy channel" model

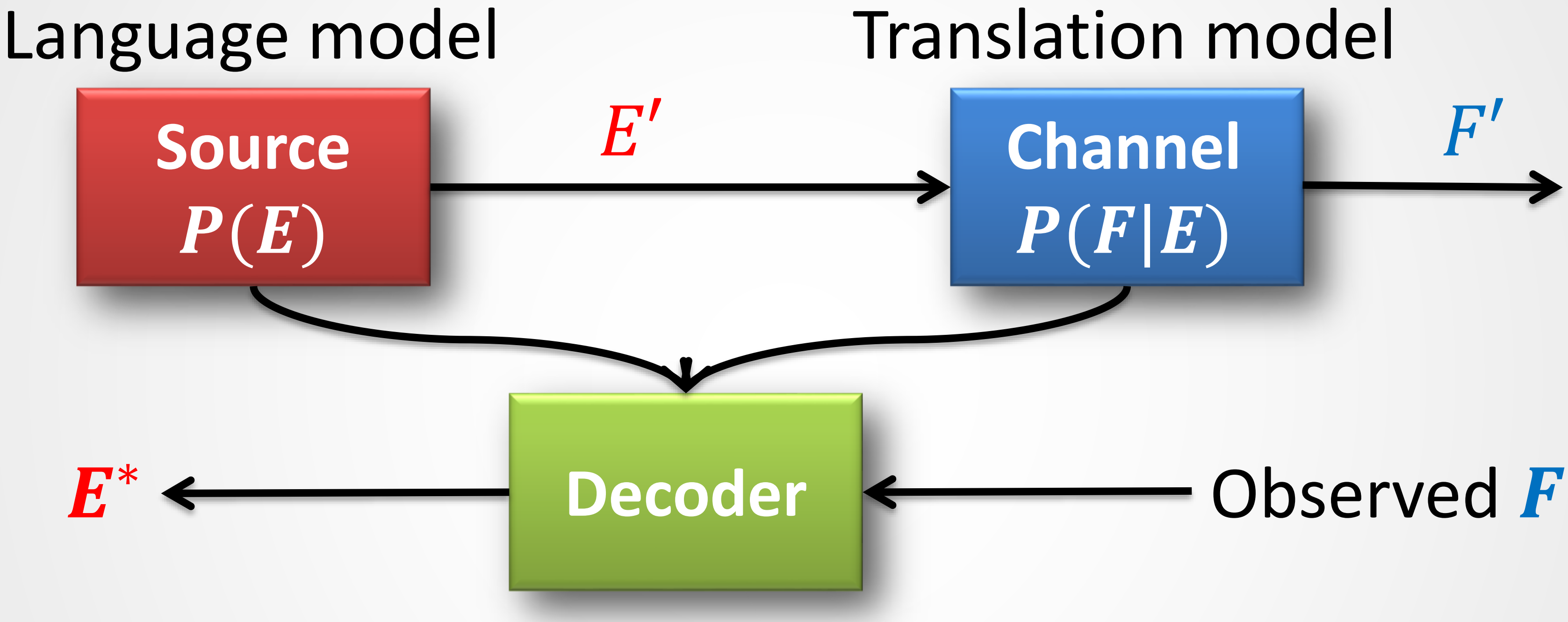
- Imagine that you're given a French sentence, F, and you want to convert it to the best corresponding English sentence, E^*
  - i.e., $E^* = \underset{E}{\operatorname{argmax}} P(E|F)$
- Use Bayes's Rule:

$$E^* = \operatorname{argmax}_E \frac{P(F|E)P(E)}{P(F)}$$

- $P(F)$ doesn't change argmax



The “noisy channel” model



$$E^* = \underset{E}{\operatorname{argmax}}\, P(F|E)P(E)$$

How SMT uses the noisy channel

- How does SMT work?

$$E^* = \underset{E}{\operatorname{argmax}} \underbrace{P(F|E)}_{\text{Translation model}} \overbrace{P(E)}^{\text{Language model}}$$

- $P(E)$ is a **language model** (e.g., *N*-gram) and encodes knowledge of word order.
- $P(F|E)$ is a **word- (or phrase-)level translation model** that encodes only knowledge on an *unordered* basis.

- **Combining** these models can give us **fluency** and **consistency**, respectively.

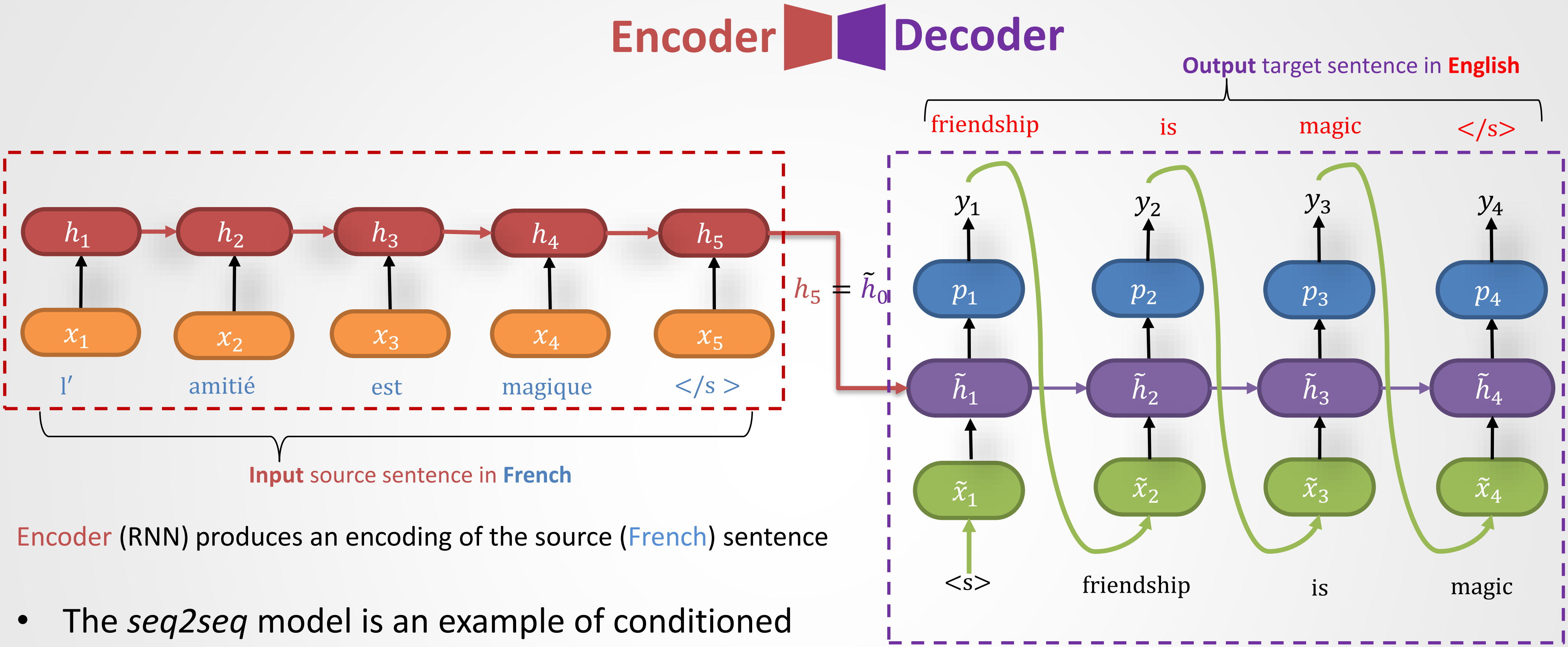
What about NMT?

- Machine translation with neural networks
- *Usually* drops noisy channel: $E^* = \text{argmax}_E P(E|F)$
  - Some NMT researchers (e.g. "Simple and effective noisy channel modeling for neural machine translation," 2019. Yee *et al.*) use an objective inspired by the noisy channel
- No (explicit) alignments – often not even sentence-aligned
- Outperforms SMT by a large margin on poorly resourced language pairs.

Solving the alignment problem

- Recall that source and target words (or, sentences) are not always one-to-one
- SMT solution is to marginalize explicit alignments
  - $E^* = \text{argmax}_E \sum_A P(F, A|E)P(E)$
- NMT uses "sequence-to-sequence (seq2seq)" encoder/decoder architectures
  - An **encoder** produces a representation of F
  - A **decoder** interprets that representation and generates an output sequence E

NMT: the seq2seq model



Encoder (RNN) produces an encoding of the source (French) sentence

Decoder (RNN) generates target sentence (in English), conditioned on the encoding

Decoder is predicting the next word of the target sentence y

Prediction is **conditioned** on the source sentence **x**

$$P(y|x) = P(y_1|x)P(y_2|y_1,x)\ldots P\big(y_T|y_1,\ldots y_{(T-1)},x\big)$$

- The *seq2seq* model is an example of conditioned language model (LM)
- Many variants exists. The classical (vanilla) seq2seq model outlined here
- NMT directly calculates $y^* = \text{argmax}_y P(y|x)$
- I.e. with our formulation:

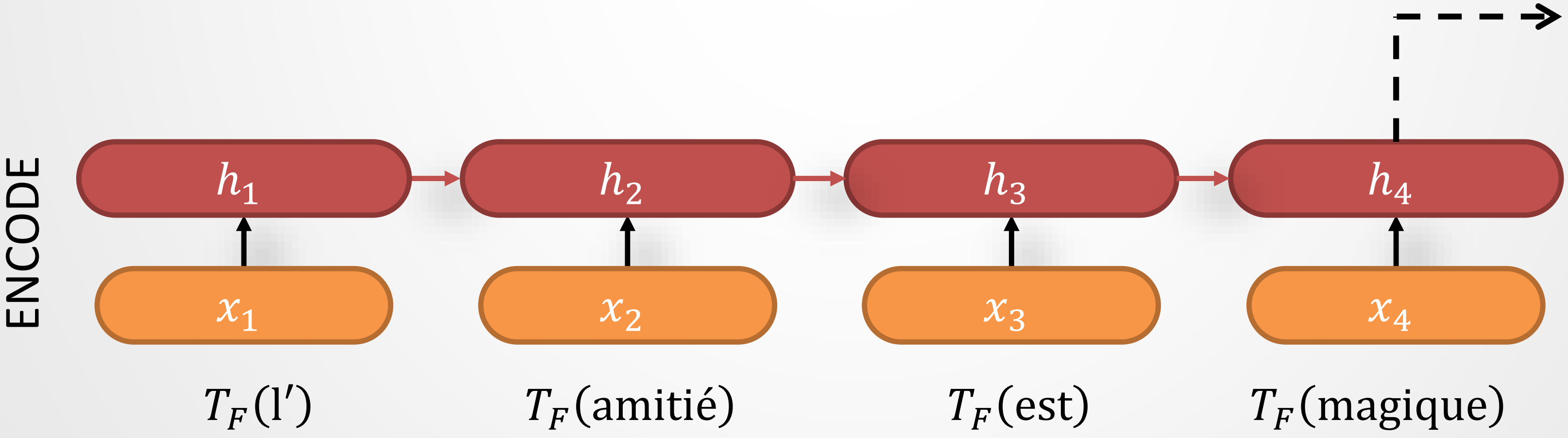
$$E^* = \text{argmax}_E P(E|F)$$

Notation

Term	Meaning
$F_{1:S}$	Source sequence (translating from)
$E_{1:T}$	Target sequence (translating to)
$x_{1:S}$	Input to encoder RNN (i.e. source embeddings $x_s = T_F(F_s)$)
$h_{1:S}^{(\ell,n)}$	Encoder hidden states (w/ optional layer index ℓ or head n)
$\tilde{x}_{1:T}$	Input to decoder RNN
$\tilde{h}_{1:T}^{(\ell,n)}$	Decoder hidden states (w/ optional layer index ℓ or head n)
$p_{1:T}$	Decoder output token distribution parameterization $p_t = f\left(\tilde{h}_t\right)$
$y_{1:T}$	Sampled output token from decoder $y_t \sim P(y_t \vert p_t)$
$c_{1:T}$	Attention context $c_t = Attend\left(\tilde{h}_t, h_{1:S}\right) = \sum_s \alpha_{t,s} h_s$
$e_{1:T,1:S}$	Score function output $e_{t,s} = score\left(\tilde{h}_t, h_s\right)$
$\alpha_{1:T,1:S}$	Attention weights $\alpha_{t,s} = \exp e_{t,s} \,/\, \sum_{s'} \exp e_{t,s'}$
$\tilde{z}_{1:T}^{(\ell)}$	Transformer decoder intermediate hidden states (after self-attention)

Encoder

- Encoder given source text $x = (x_1, x_2, \ldots)$
  - $x_s = T_F(F_s)$ a source word embedding
- Outputs last hidden state of RNN
- Note $h_S = f(F_{1:S})$ conditions on entire source



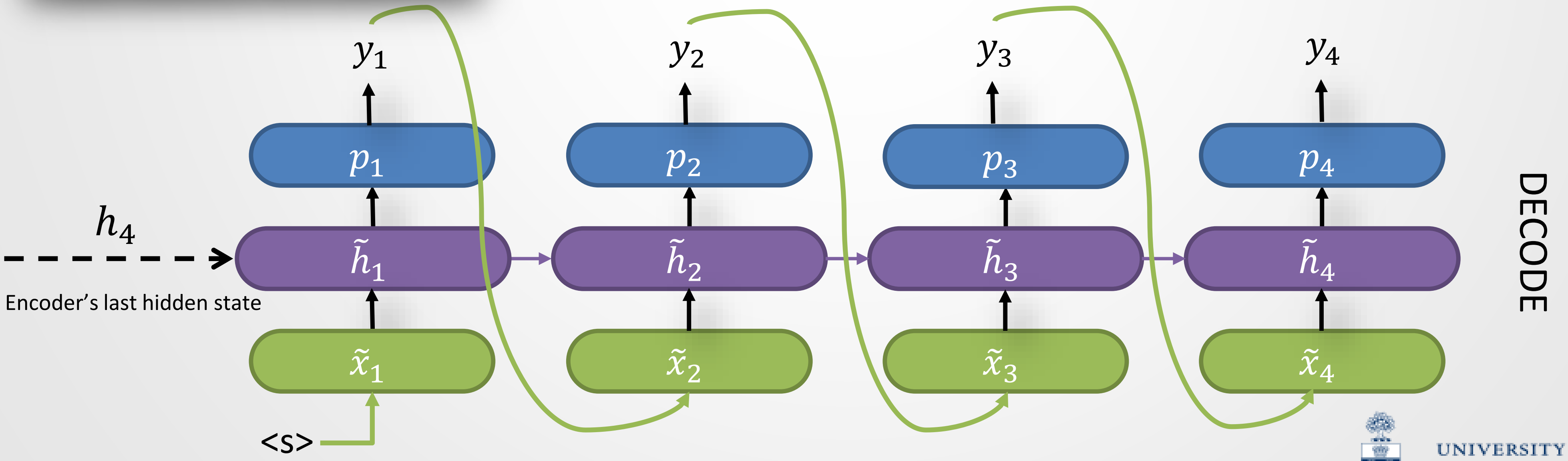
Source sentence (French): *L' amitié est magique*

Target sentence (English): *Friendship is magic* [Ground truth output]

Decoder

- **Sample** a target sentence word by word $y_t \sim P(y_t|p_t)$
- Set input to be embedding of **previously generated word** $\tilde{x}_t = T_E(\boldsymbol{y_{t-1}})$
- $p_t = f(\tilde{h}_t) = f\left(g(\tilde{x}_t, \tilde{h}_{t-1})\right)$ is **deterministic**
- Base case: $\tilde{x}_1 = T_E(\text{<s>}),\ \tilde{h}_0 = h_S$
- $P(y_{1:T}|F_{1:S}) = \prod_t P(y_t|y_{<t}, F_{1:S})$ → **auto-regressive**

N.B.: Implicit $y_0 = \text{<s>}, P(y_0) = 1$



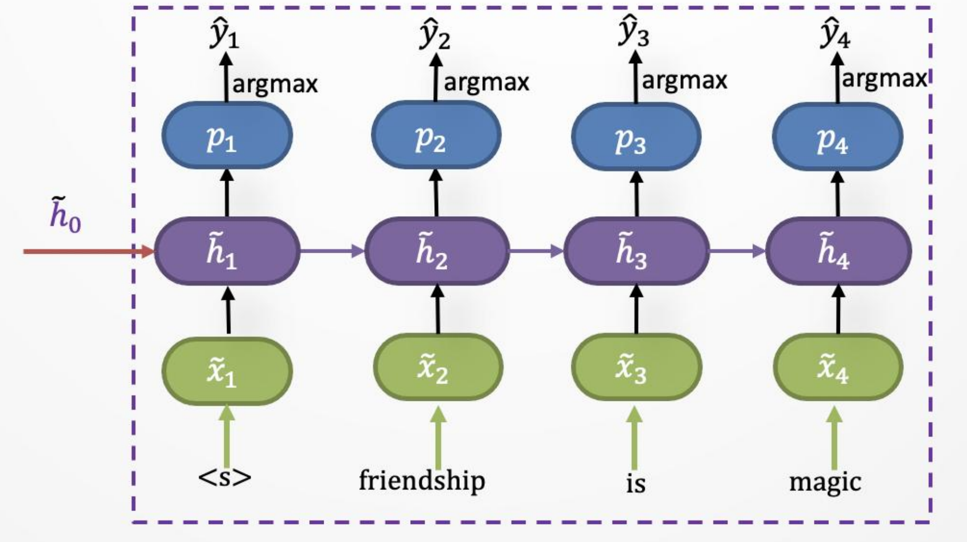
NMT: Training a MT system

- Train towards maximum likelihood estimate (MLE) against **one** translation E
- Auto-regression simplifies independence

MLE: $\theta^* = \operatorname{argmin}_\theta \mathcal{L}(\theta|E,F)$ $\quad \mathcal{L}(\theta|E,F) = -\log P_\theta(y = E|F)$

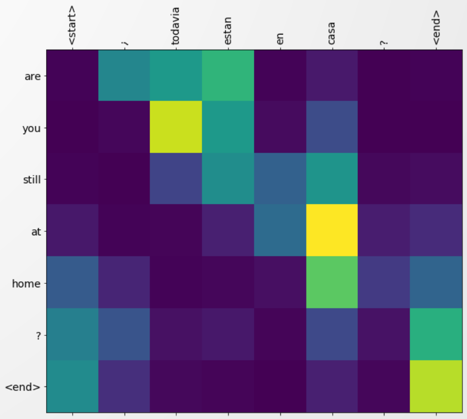
$$= -\sum_t \log P_\theta(y_t = E_t|E_{<t}, F_{1:S})$$

$$\mathcal{L} = -\log P(\text{friendship}|\cdots) - \log P(\text{is}|\cdots) - \log P(\text{magic}|\cdots) - \log P(\text{</s>}|\cdots)$$



Attention advantages

- Improves NMT performance significantly (reply to RNN)
- Appears to solve the bottleneck problem
  - Allows the decoder to look at the source sentence directly, circumventing the bottleneck
- Helps with the long-horizon (vanishing gradient) problem – by providing shortcut to distant states
- Makes the model (somewhat) interpretable
  - We can examine the attention distribution to see what the decoder was focusing on
- We get soft alignment for free
  - Compare w/ the '*word alignment*' matrix from SMT
  - This was also often soft
  - Comes from only sentence-aligned input
  - There had already been a number of unsupervised alignment methods proposed for SMT


<start>
¿
todavia
estan
en
casa
?
<end>
are
you
still
at
home
?
<end>