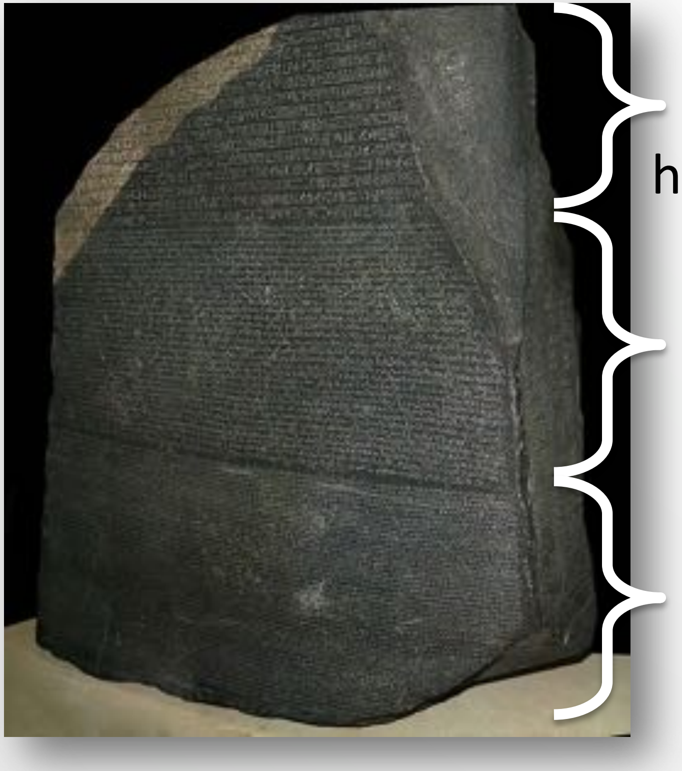
Statistical machine translation

CSC401/2511 – Natural Language Computing – Spring 2021
Lecture 6 Frank Rudzicz, Sean Robertson, Serena Jeblee
University of Toronto

The Rosetta Stone

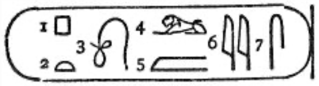
- The **Rosetta Stone** dates from 196 BCE.
  - It was re-discovered by French soldiers during Napoleon’s invasion of Egypt in 1799 CE.

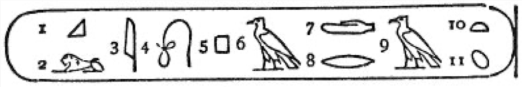


- It contains three **parallel** texts in different languages, only the **last** of which was understood.

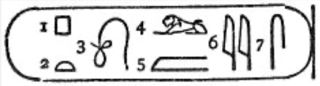
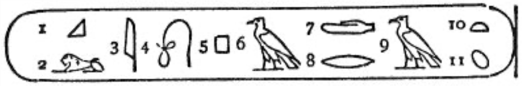
- *By 1799, ancient Egyptian had been forgotten.*

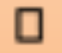
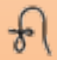
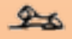
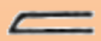
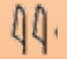
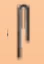
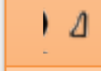
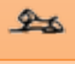
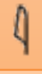
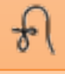
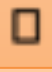
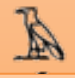
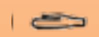
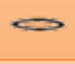
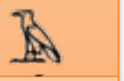
Deciphering Rosetta

- During 1822–1824, **Jean-François Champollion** worked on the Rosetta stone. He noticed:
  1. The circled Egyptian symbols  appeared in roughly the same positions as the word '*Ptolemy*' in the Greek.

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

Aside – deciphering Rosetta

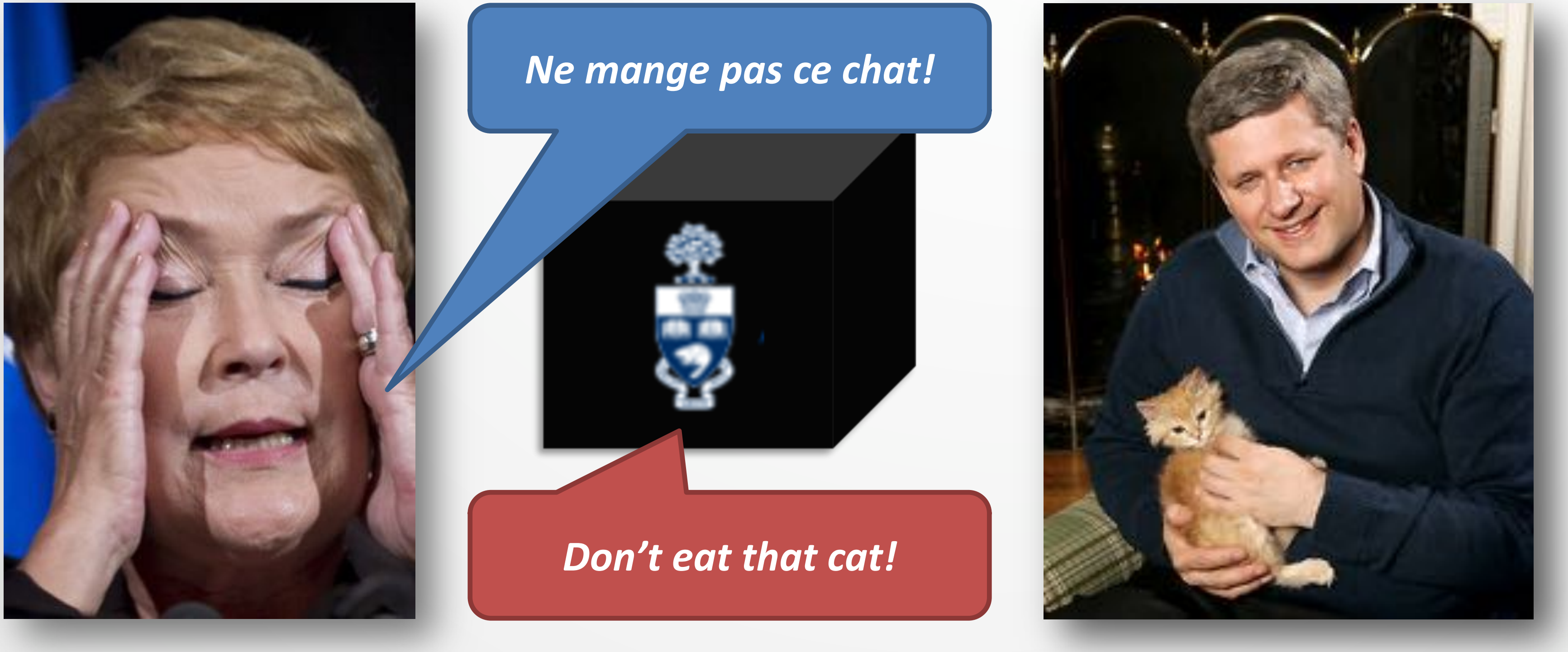
- So if  was '*Ptolemy*' and  was '*Cleopatra*' and the symbols corresponded to sounds – can we match up the symbols?

								
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:
  - *It would not have been possible without* **aligning** *unknown words (hieroglyhs) to known words (Greek)…*

Today

- Introduction to statistical machine translation (SMT).
    - What we want is a system to take utterances/sentences in one language and transform them to another:



Ne mange pas ce chat!

Don’t eat that cat!

Direct translation

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

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

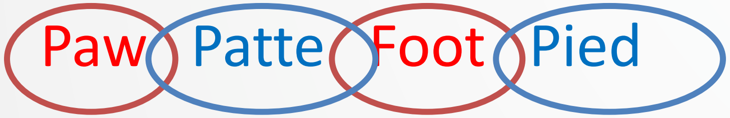
Mi	*nombre*	*es*	*T-bone*
My	*name*	*is*	*T-bone*
Mon	*nom*	*est*	*T-bone*

Difficulties in MT: typology

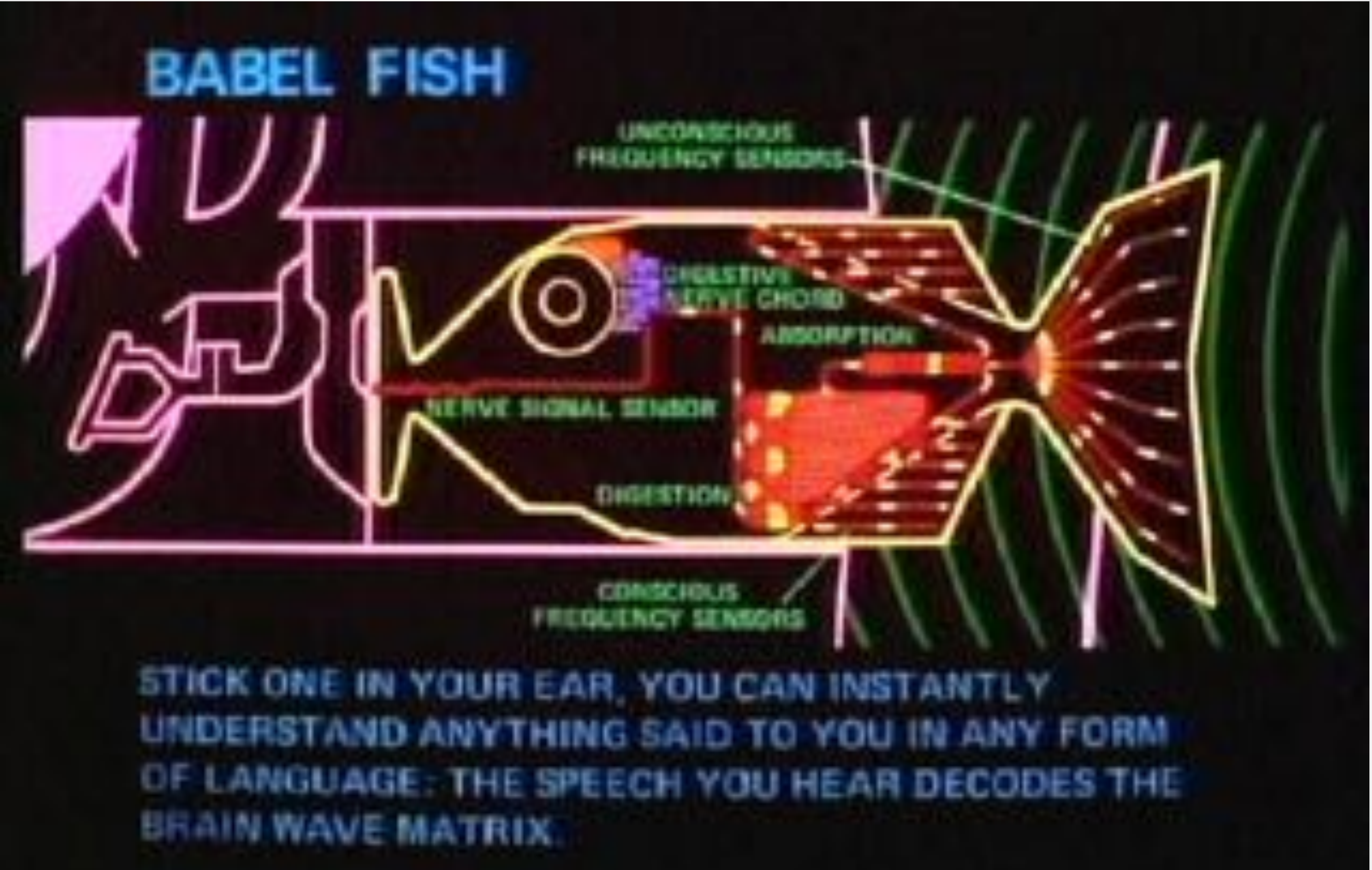
- Different morphology → difficult mappings, *e.g.*
  - Many (*polysynthetic*) vs one (*isolating*) morphemes per word
  - Many (*fusion*) vs few (*agglutinative*) features per morpheme
- Different syntax → long-distance effects, *e.g.*
  - SVO vs. SOV vs. VSO (e.g. English vs. Japanese vs. Arabic)
    - He listens to music / kare ha ongaku wo kiku
  - Verb- vs. satellite-framed (e.g. Spanish vs. English)
    - La botella salió flotando / The bottle floated out

Difficulties in MT: ambiguity

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



  - 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


BABEL FISH
UNCONSCIOUS FREQUENCY SENSORS
DIGESTIVE NERVE CHORD
ABSORPTION
NERVE SIGNAL SENSOR
DIGESTION
CONSCIOUS FREQUENCY SENSORS
STICK ONE IN YOUR EAR, YOU CAN INSTANTLY UNDERSTAND ANYTHING SAID TO YOU IN ANY FORM OF LANGUAGE: THE SPEECH YOU HEAR DECODES THE BRAIN WAVE MATRIX.

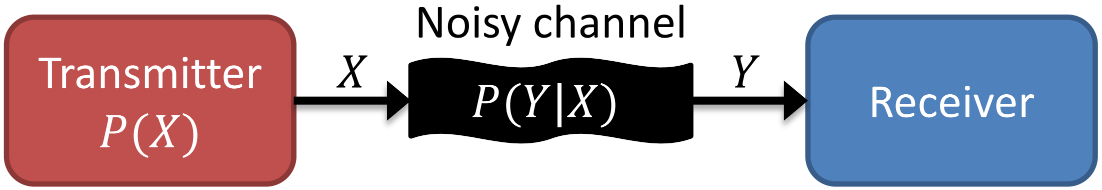
THE NOISY CHANNEL

Statistical machine translation

- Machine translation seemed to be an intractable problem until a change in perspective…

> When I look at an article in Russian, I say: ‘This is really written in English, but it has been **coded** in some strange symbols. I will now proceed to **decode**.’

Warren Weaver March, 1947



Claude Shannon July, 1948

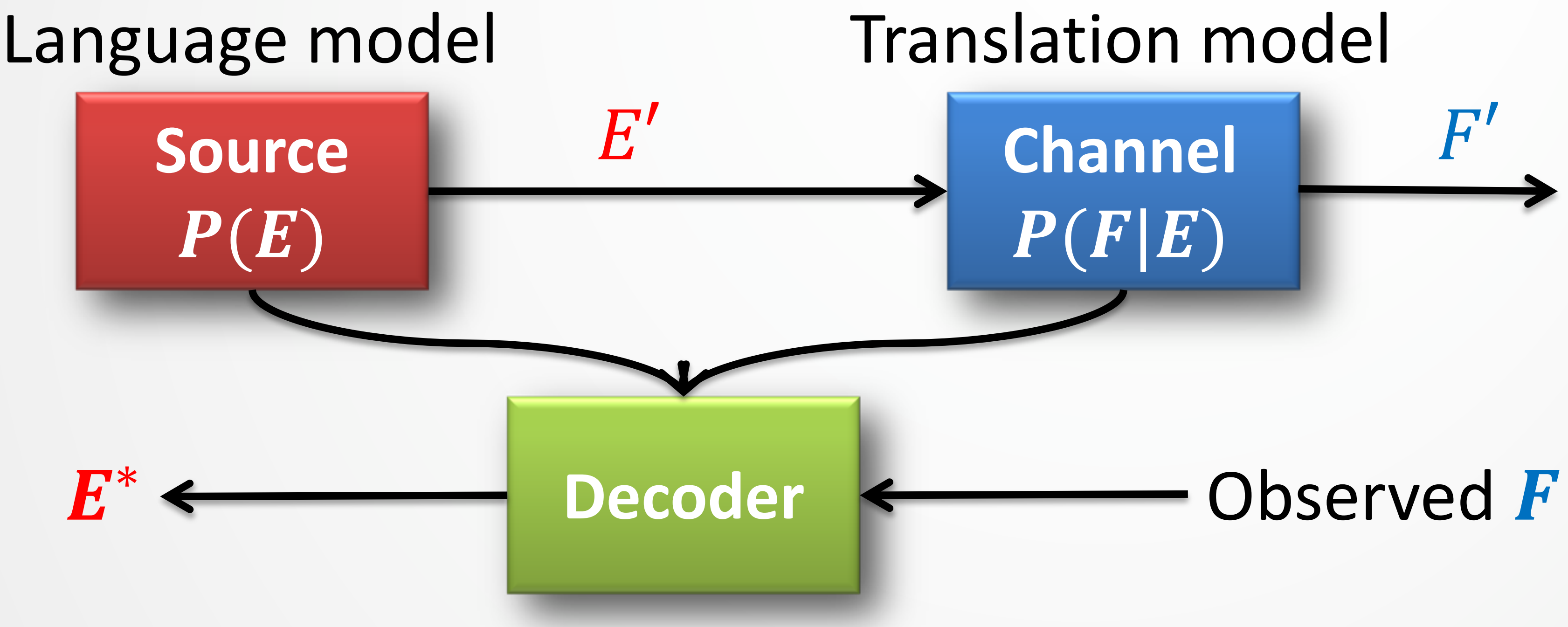
The noisy channel model

- 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' Rule:

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

- $P(F)$ doesn't change argmax (besides, French isn't anything but noisy English anyway)

The noisy channel



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

How to use the noisy channel

- How does this work?

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

- $P(E)$ 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 **naturalness** and **fidelity**, respectively.

How to use the noisy channel

- Example from Koehn and Knight using only conditional likelihoods of Spanish words given English words.

- *Que hambre tengo yo*

  →

What hunger have I	$P(S\|E) = 1.4E^{-5}$
Hungry I am so	$P(S\|E) = 1.0E^{-6}$
I am so hungry	$P(S\|E) = 1.0E^{-6}$
Have I that hunger	$P(S\|E) = 2.0E^{-5}$

…

How to use the noisy channel

- … and with the English language model

- *Que hambre tengo yo*

  →

What hunger have I	$P(S\|E)P(E) = 1.4E^{-5} \times 1.0E^{-6}$
Hungry I am so	$P(S\|E)P(E) = 1.0E^{-6} \times 1.4E^{-6}$
I am so hungry	$P(S\|E)P(E) = 1.0E^{-6} \times 1.0E^{-4}$
Have I that hunger	$P(S\|E)P(E) = 2.0E^{-5} \times 9.8E^{-7}$

…

How to learn $P(F|E)$?

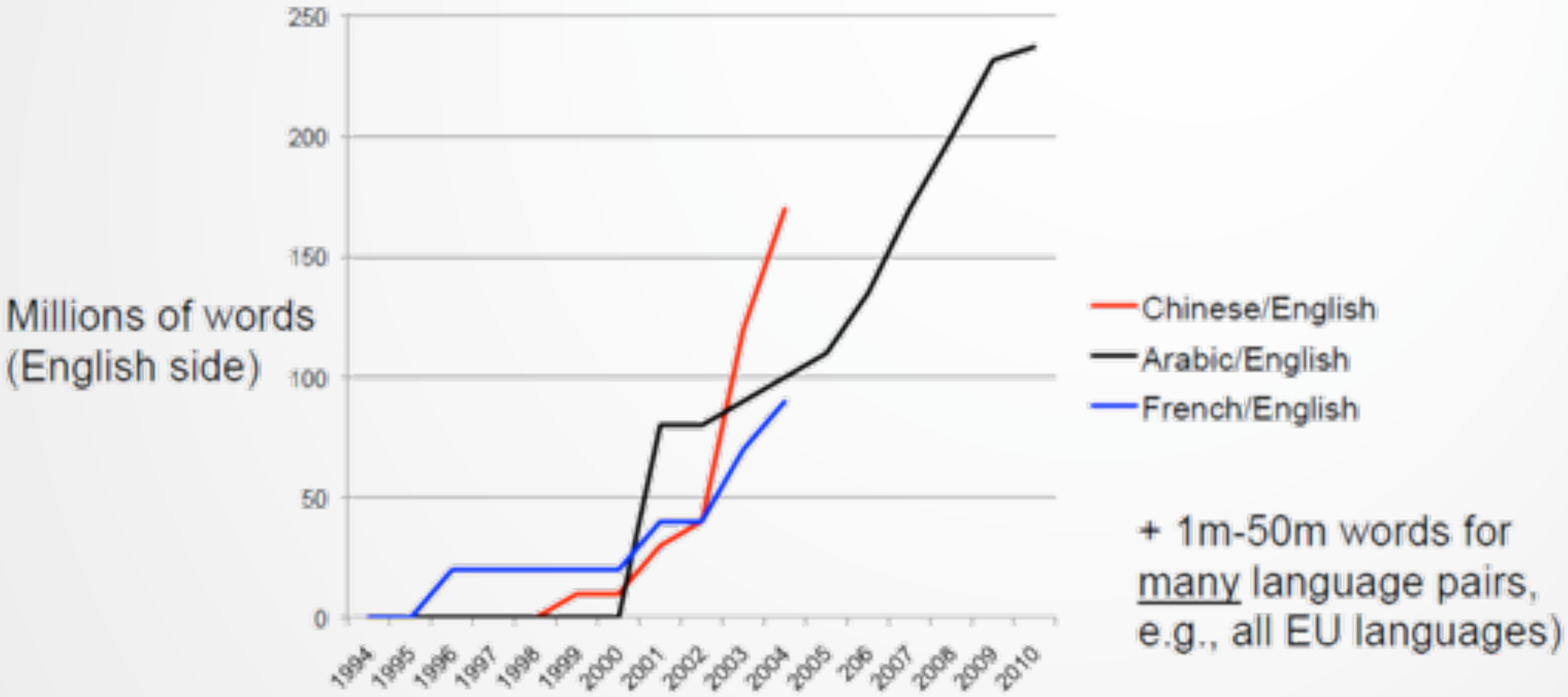
- Solution: collect statistics on vast parallel texts

… **citizen** of Canada has the **right** to vote in an election of members of the House of Commons or of a legislative assembly and to be qualified for membership …

… **citoyen** canadien a le **droit** de vote et est éligible aux élections législatives fédérales ou provinciales …

e.g., the *Canadian Hansards*:
bilingual Parliamentary proceedings

Bilingual data



+ 1m-50m words for <u>many</u> language pairs, e.g., all EU languages)

From Chris Manning's course at Stanford

- Data from Linguistic Data Consortium at University of Pennsylvania.

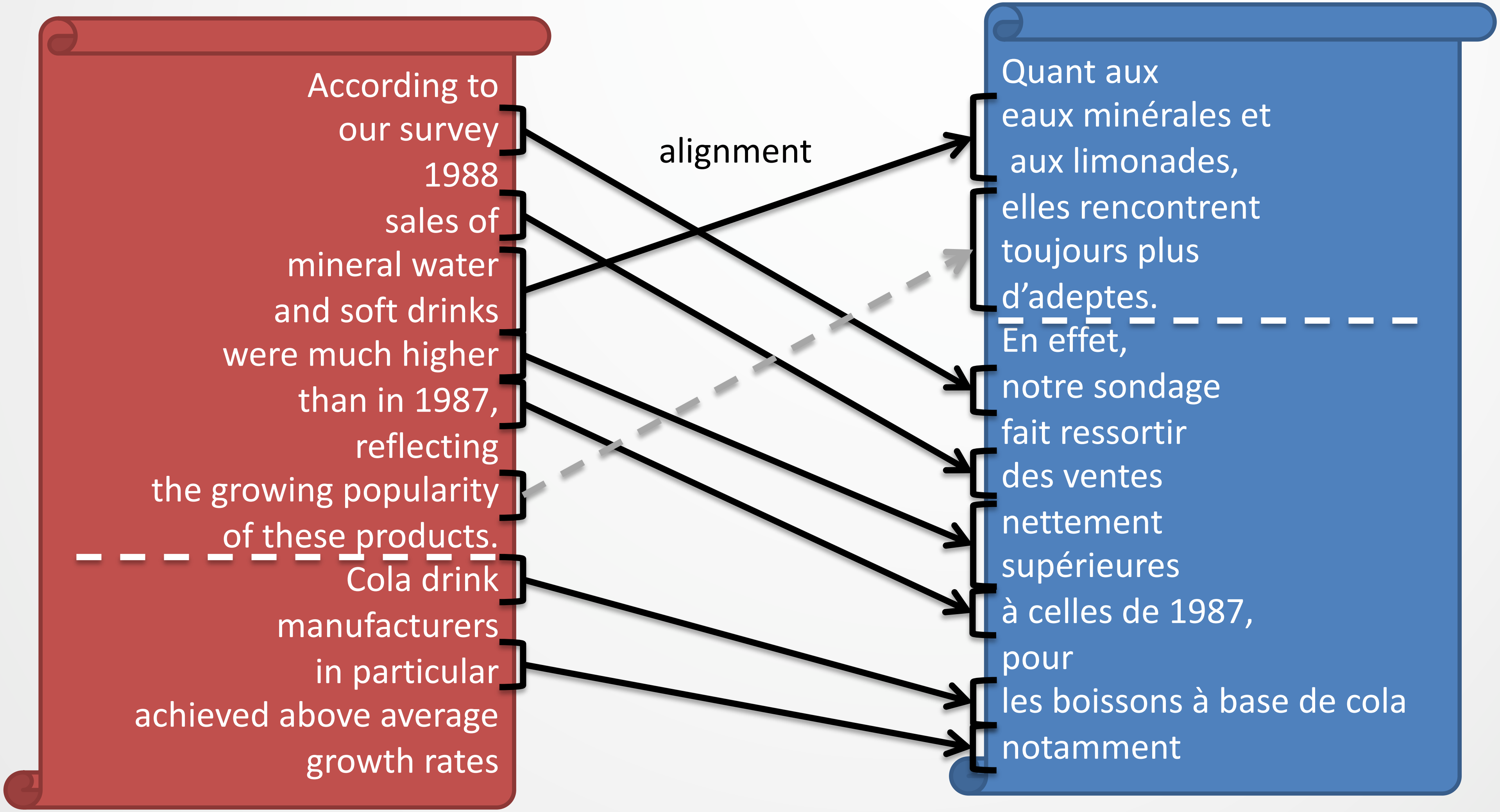
Alignments

- Alignments at different granularities
  - Word, phrase, sentence, document
- SMT makes alignments explicit
  - One block of text entirely responsible for a translated block (conditional independence)
- Letting A index pairs of aligned blocks in bitext

$$P(F|E) = \sum_A P(F, A|E) = \sum_A P(A|E) \prod_i P\left(F_{A_{i,1}} \middle| E_{A_{i,2}}\right)$$

Alignment

- In practice, words and phrases can be out of order.



From Manning & Schütze

Alignment

- Also in practice, we’re usually not given the alignment.

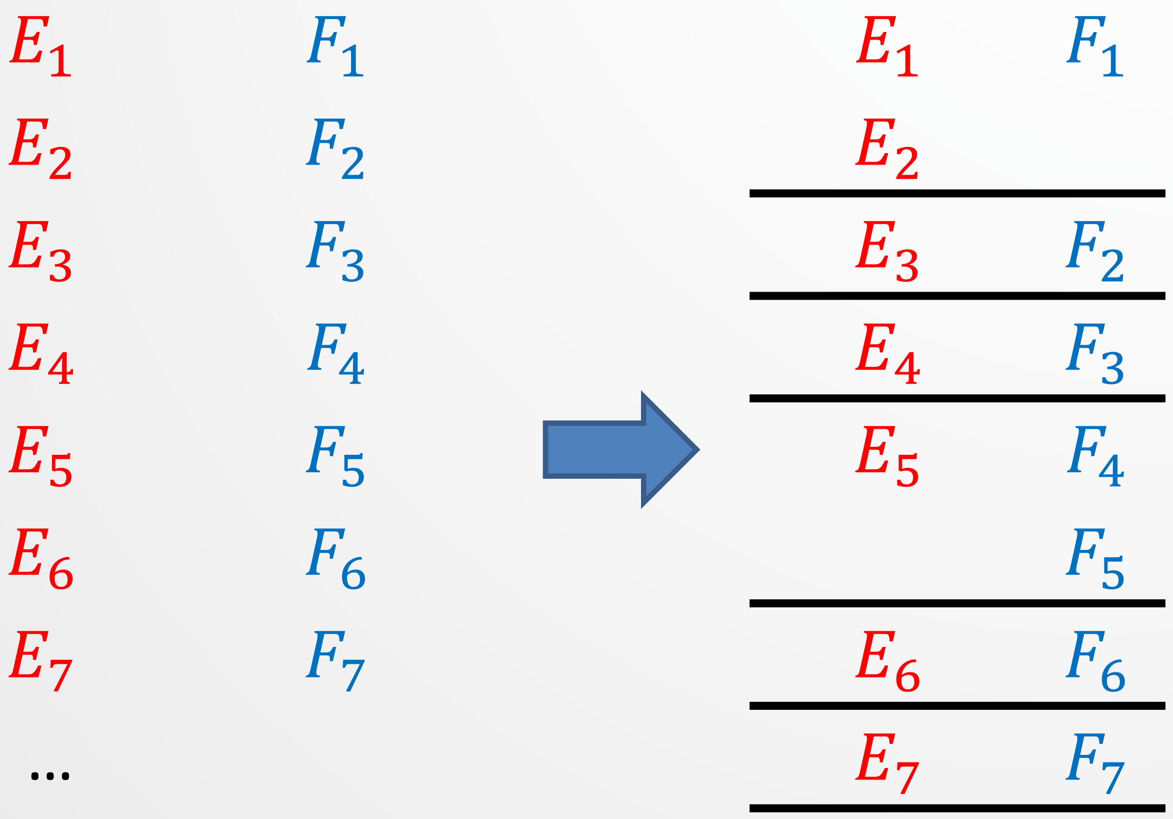
According to
our survey
1988
sales of
mineral water
and soft drinks
were much higher
than in 1987,
reflecting
the growing popularity
of these products.
Cola drink
manufacturers
in particular
achieved above average
growth rates

Quant aux
eaux minérales et
aux limonades,
elles rencontrent
toujours plus
d’adeptes.
En effet,
notre sondage
fait ressortir
des ventes
nettement
supérieures
à celles de 1987,
pour
les boissons à base de cola
notamment

From Manning & Schütze

Sentence alignment

- Sentences can also be **unaligned** across translations.
    - E.g., *He was happy.*$_{E1}$ *He had bacon.*$_{E2}$ → *Il était heureux parce qu'il avait du bacon.*$_{F1}$



E_1	F_1
E_2	F_2
E_3	F_3
E_4	F_4
E_5	F_5
E_6	F_6
E_7	F_7
…	

E_1 E_2	F_1
E_3	F_2
E_4	F_3
E_5	F_4 F_5
E_6	F_6
E_7	F_7
…	

Recalling
$\prod_i P\left(F_{A_{i,1}} \middle| E_{A_{i,2}}\right)$:
$A_1 = (\{1\}, \{1,2\})$
$A_2 = (\{2\}, \{3\})$
$A_3 = (\{4\}, \{3\})$
$A_4 = (\{4,5\}, \{5\})$
Etc…

Sentence alignment

- We often need to align **sentences** before moving forward.
- Goal: find $A^* = \text{argmax}_A P(A|F, E)$
- We'll look at two broad classes of methods:
  1. Methods that only look at **sentence length**,
  2. Methods based on **lexical matches**, or "cognates".
- Most MT (including neural) relies on sentence-level alignments of bitexts

1. Sentence alignment by length

(Gale and Church, 1993)

- **Idea**: lengths of aligned sentences are correlated
- Assuming the paragraph alignment is known,
  - $\mathcal{L}_E$ is the # of characters in an English sentence,
  - $\mathcal{L}_F$ is the # of characters in a French sentence.
- Define cost/penalty function $Cost(\mathcal{L}_E, \mathcal{L}_F)$
  - Lowest when $\mathcal{L}_E = c\mathcal{L}_F$ for learned/guessed c
- Also define "prior" fixed cost $C_{i,j}$ of aligning i English sentences to j French sentences

1. Sentence alignment by length

E_1 E_2	F_1
E_3	F_2
E_4	F_3
E_5	F_4 F_5
E_6	F_6

$$
\begin{aligned}
Cost = {} & Cost\left(\mathcal{L}_{E_1} + \mathcal{L}_{E_2}, \mathcal{L}_{F_1}\right) + C_{2,1} + \\
& Cost\left(\mathcal{L}_{E_3}, \mathcal{L}_{F_2}\right) + C_{1,1} + \\
& Cost\left(\mathcal{L}_{E_4}, \mathcal{L}_{F_3}\right) + C_{1,1} + \\
& Cost\left(\mathcal{L}_{E_5}, \mathcal{L}_{F_4} + \mathcal{L}_{F_5}\right) + C_{1,2} + \\
& Cost\left(\mathcal{L}_{E_6}, \mathcal{L}_{F_6}\right) + C_{1,1}
\end{aligned}
$$

Find distribution of sentence breaks with minimum cost using **dynamic programming**

It's a bit more complicated – see paper on course webpage (**aside**)

2. Sentence alignment by cognates

- **Cognates**: *n.pl.* Words that have a common **etymological** origin.
- **Etymological**: *adj.* Pertaining to the historical derivation of a word. E.g., *porc*→*pork*

- The intuition is that words that are **related** across languages have similar **spellings**.
  - e.g., *zombie*/*zombie*, *government*/*gouvernement*
  - Not always: *son* (male offspring) vs. *son* (sound)

- Cognates can "anchor" sentence alignments between related languages.

2. Sentence alignment by cognates

- Cognates should be spelled similarly…

- ***N*-graph**: *n.* Similar to *N*-grams, but computed at the **character-level**, rather than at the word-level.

  E.g., $Count(s, h, i)$ is a **trigraph** model

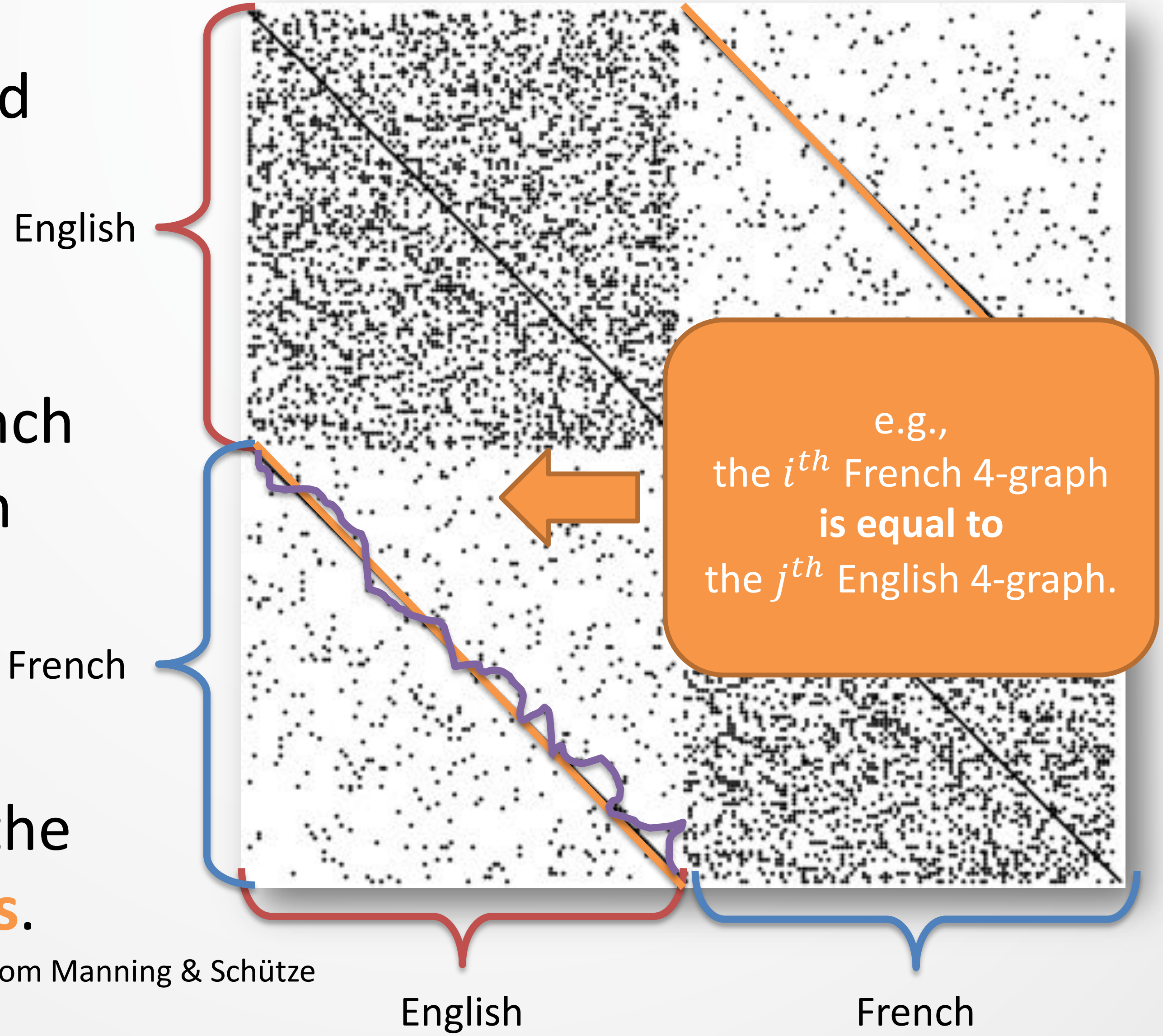
- Church (1993) tracks all **4-graphs** which are identical across two texts.
  - He calls this a ‘signal-based’ approximation to cognate identification.
  - Better for noisy data, like the results of optical character recognition

2. Church's method

1. Concatenate paired texts.

2. Place a 'dot' where the i^{th} French and the j^{th} English 4-graph are **equal**.

3. Search for a **short path** 'near' the **bilingual diagonals**.

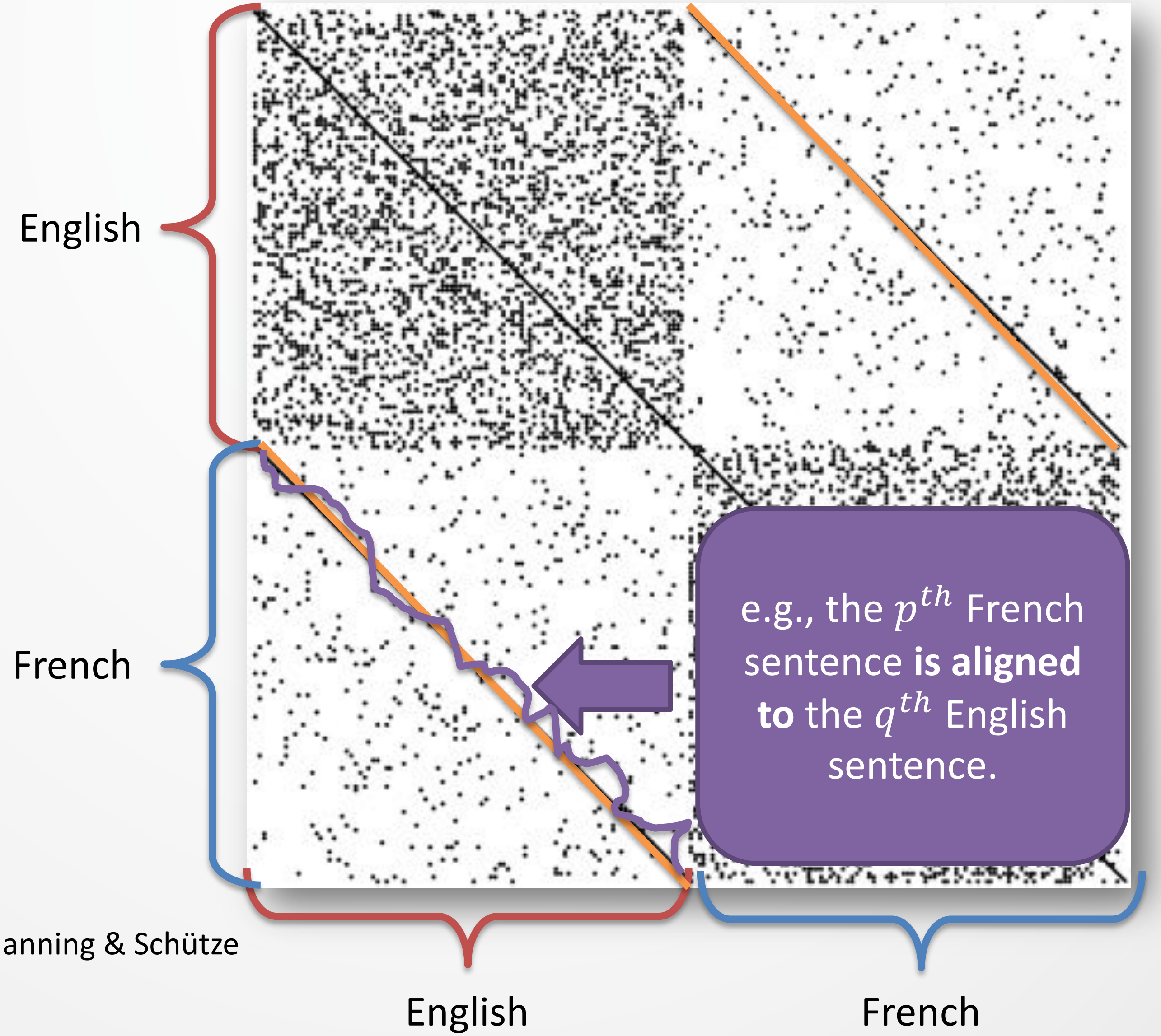


e.g., the i^{th} French 4-graph **is equal to** the j^{th} English 4-graph.

From Manning & Schütze

2. Church's method

- Each point along **this path** is considered to represent a **match** between languages.
- The relevant English and French sentences are ∴ **aligned**.



e.g., the p^{th} French sentence **is aligned to** the q^{th} English sentence.

From Manning & Schütze

Aligning other granularities

- Recall: $P(F|E) = \sum_A P(A|E) \prod_i P\left(F_{A_{i,1}} \middle| E_{A_{i,2}}\right)$

- A_i can be pairs of sets of sentences if E, F are documents

- If E, F are sentences, A_i are pairs of sets of words

Word alignment models

- Make a simplifying assumption that every word in F maps to one E (i.e. $A_i = (\{i\}, \{j\}) \mapsto j$)
- E.g. IBM-1: $P(F|A,E) \propto \prod_i P\left(F_i \middle| E_{A_i}\right)$

$$\frac{Count\left(F_i, E_{A_i}\right)}{Count\left(E_{A_i}\right)}$$

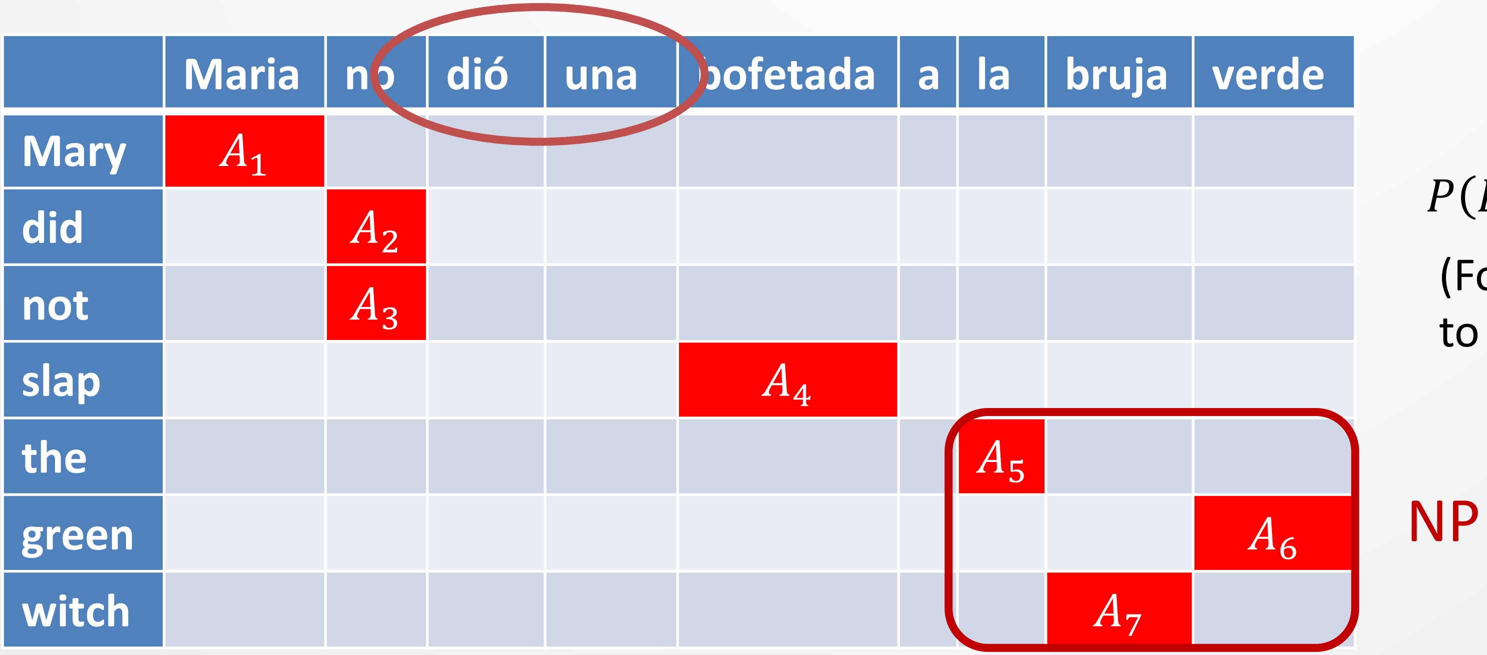
- Trained via Expectation Maximization (see HMM lecture)

	Maria	no	dió	una	bofetada	a	la	bruja	verde
Mary	A_1								
did						A_6			
not		A_2							
slap			A_3	A_4	A_5				
the							A_7		
green									A_9
witch								A_8	

From J&M 2nd Ed.

Problems with word alignments

- What if some E_j isn't aligned anywhere?
- Need more flexible context!



	Maria	no	dió	una	bofetada	a	la	bruja	verde
Mary	A_1								
did		A_2							
not		A_3							
slap					A_4				
the							A_5		
green									A_6
witch								A_7	

$P(E|F)$

(For English to Spanish)

Phrase-based translation

- Suppose beads are pairs non-empty, contiguous spans of words that partition $F \times E$

$$A_i = \left(\left(\ell_1^{(i)} : u_1^{(i)}\right), \left(\ell_2^{(i)} : u_2^{(i)}\right)\right)$$

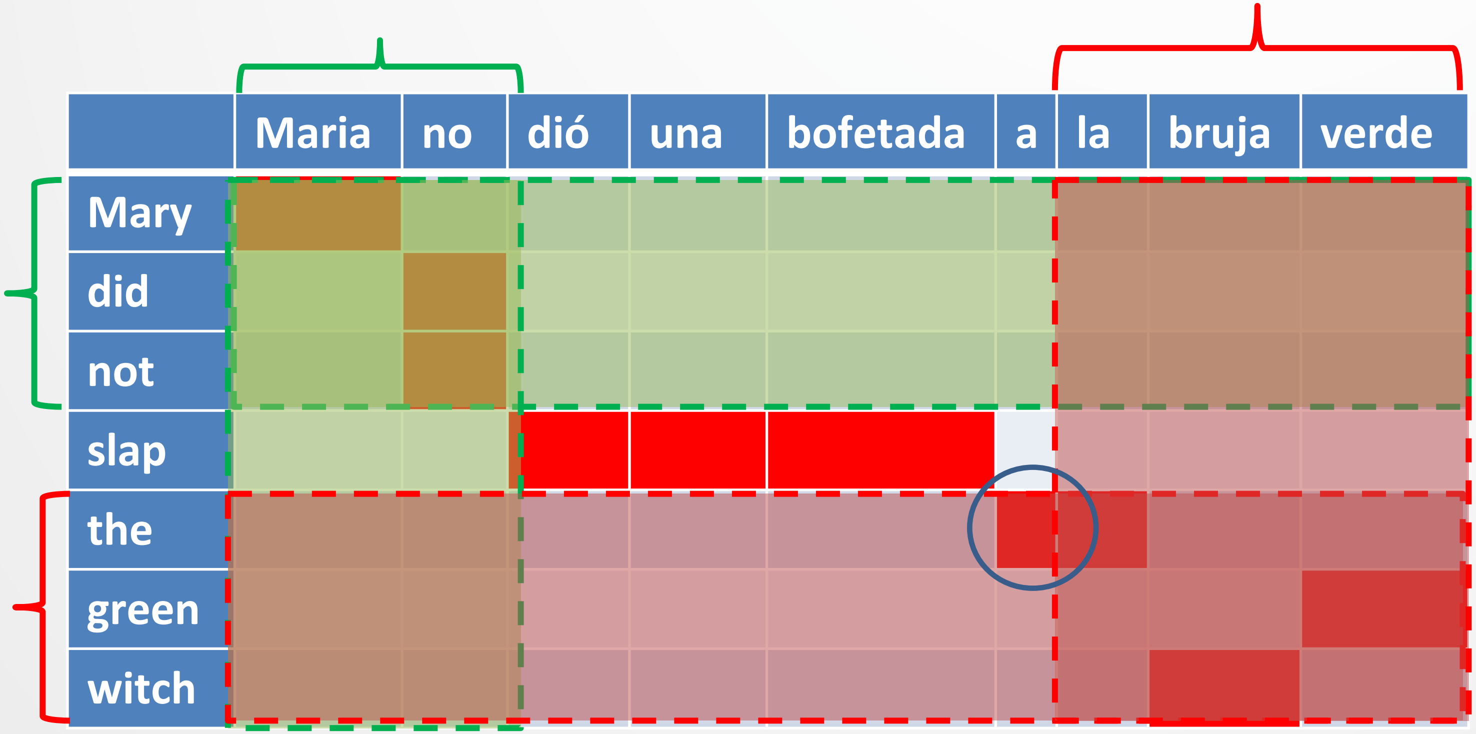
- Call each span an indivisible phrase $\left(F_{A_{i,1}}, E_{A_{i,2}}\right) \mapsto (\bar{F}_i, \bar{E}_i)$ and assume phrases sequential in E, then:

$$P(F, A|E) \propto \prod_i \phi(\bar{F}_i, \bar{E}_i) d\left(u_1^{(i-1)} - \ell_1^{(i)} - 1\right)$$

- $d(\cdot)$ is the distortion model/distance (e.g. $d(x) = \alpha^{|x|}$)
  - Since $\bar{E}_i, \bar{E}_{i+1}$ are sequential, penalizes when $\bar{F}_i, \bar{F}_{i+1}$ aren't
- $\phi(\bar{F}, \bar{E}) = Count(\bar{F}, \bar{E}) / \sum_{\bar{F}'} Count(\bar{F}', \bar{E})$ is the phrase translation probability

Bilingual phrase pairs

- Count the pair $(\bar{F}, \bar{E}) = \left(F_{\ell_1:u_1}, E_{\ell_2:u_2}\right)$ if “consistent”
  1. At least one A_i is in the box $[\ell_1 : u_1] \times [\ell_2 : u_2]$
  2. All A_i containing any word in $[\ell_1 : u_1]$ or any word in $[\ell_2 : u_2]$ must be in the box as well



	Maria	no	dió	una	bofetada	a	la	bruja	verde
Mary	■								
did		■							
not		■							
slap			■	■	■				
the						■	■		
green									■
witch								■	

Decoding with phrases

- Decoding is the process of deriving E given F
  $E^* = \text{argmax}_E P(F|E)P(E) \approx \text{argmax}_E P(F,A|E)P(E)$
- Checking all E, A is infeasible
- Instead, use a (heuristic) **beam search**
  1. Choose partial translation (E', A') with highest score ($\propto P(F', A'|E')P(E')$)
  2. Increment that by appending bilingual phrase pairs
  3. Prune set of resulting partial translations by score
- We'll see beam search in more detail in NMT

NEURAL MACHINE TRANSL-ATION

What is 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 the noisy channel objective
- No (explicit) alignments
- Outperforms "SMT" by a large margin

Solving the alignment problem

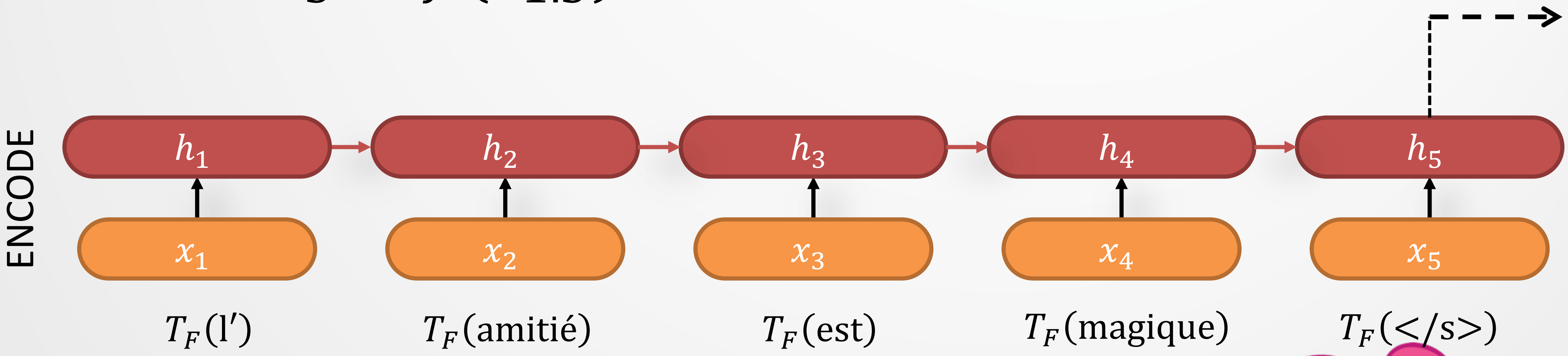
- Recall that source and target words (/sentences) are not always one-to-one
- SMT solution is to marginalize explicit alignments $E^* = \mathrm{argmax}_E \sum_A P(F, A|E)P(E)$
- NMT uses sequence-to-sequence (seq2seq) encoder/decoder architectures
  - An **encoder** produces a representation of F
 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

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 $y_t \sim P(y_t \mid p_t)$
$c_{1:T}$	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

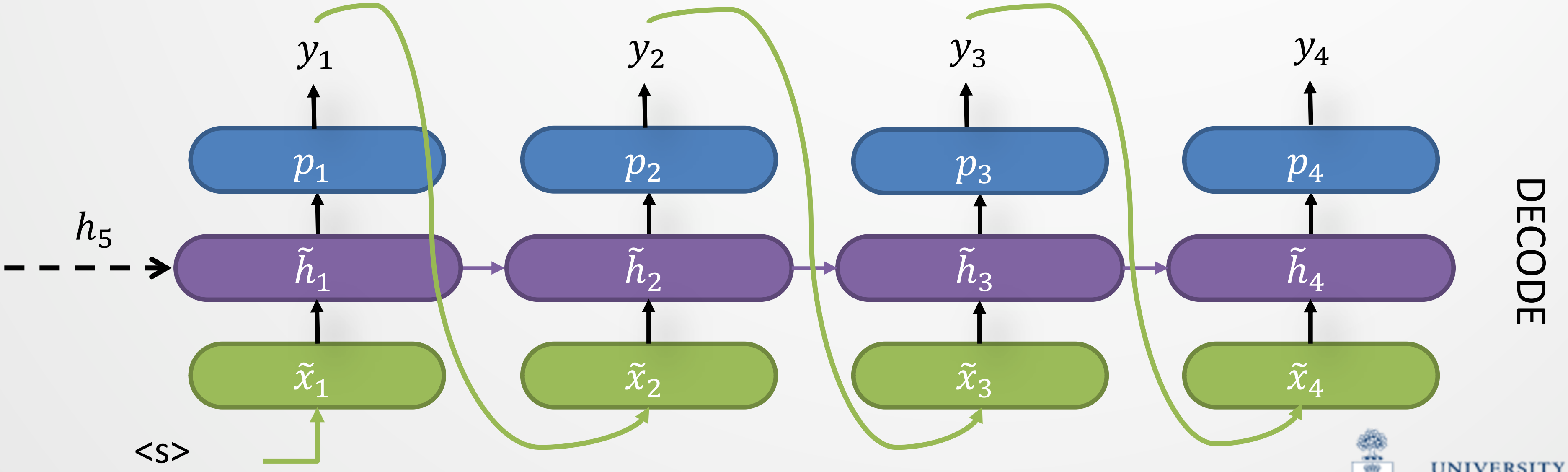
- Encoder given source text $x = (x_1, x_2, \ldots)$
  - $x_s = T_F(F_s)$ a source word embedding
- Outputs last hidden state of RNN
- Note $h_S = f(F_{1:S})$ conditions on entire source



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** $\tilde{x}_t = T_E(y_{t-1})$
- $p_t = f(\tilde{h}_t) = f\left(g(\tilde{x}_t, \tilde{h}_{t-1})\right)$ 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 $y_0 = \text{<s>}, P(y_0) = 1$



Training

- Train towards maximum likelihood estimate against **one** translation E
- Auto-regression simplifies independence
- MLE: $\theta^* = \operatorname{argmin}_\theta \mathcal{L}(\theta|E,F)$

$$\mathcal{L}(\theta|E,F) = -\log P_\theta(y = E|F)$$

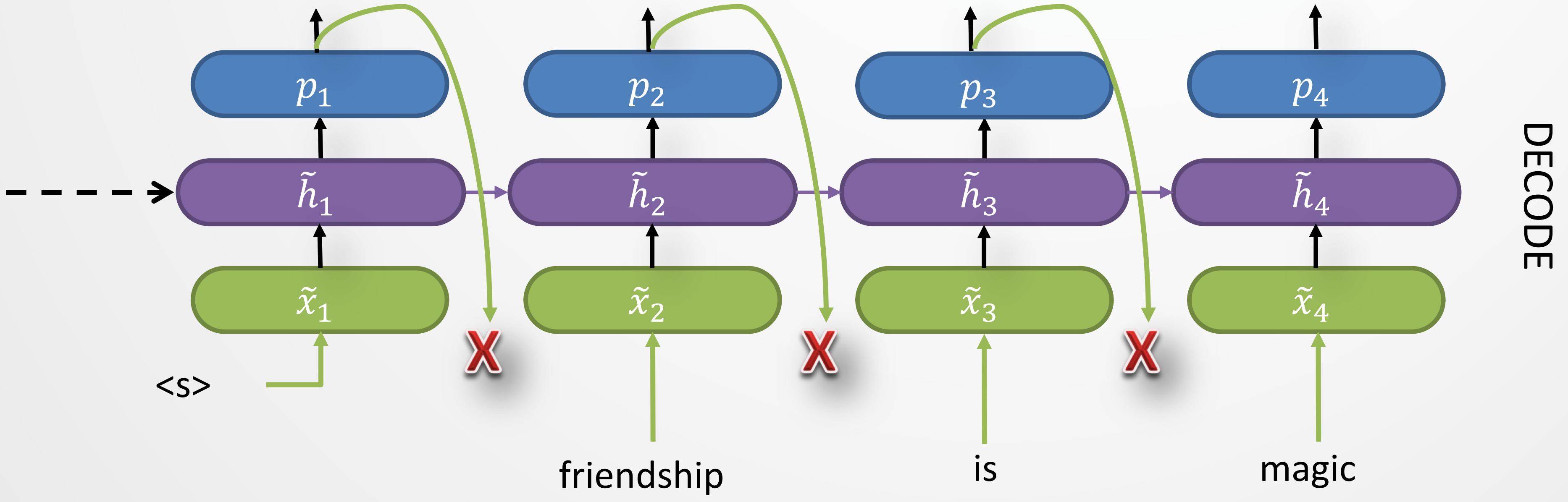
$$= -\sum_t \log P_\theta(y_t = E_t|E_{<t}, F_{1:S})$$

- Expectation maximization marginalizes over unobserved variables (e.g. alignments), **this doesn't**

Teacher forcing

- Teacher forcing = maximum likelihood estimate
- Replace $\tilde{x}_t = T(y_{t-1})$ with $\tilde{x}_t = T(E_{t-1})$
- Since $y_{t-1} \neq E_{t-1}$ in general, causes **exposure bias**

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

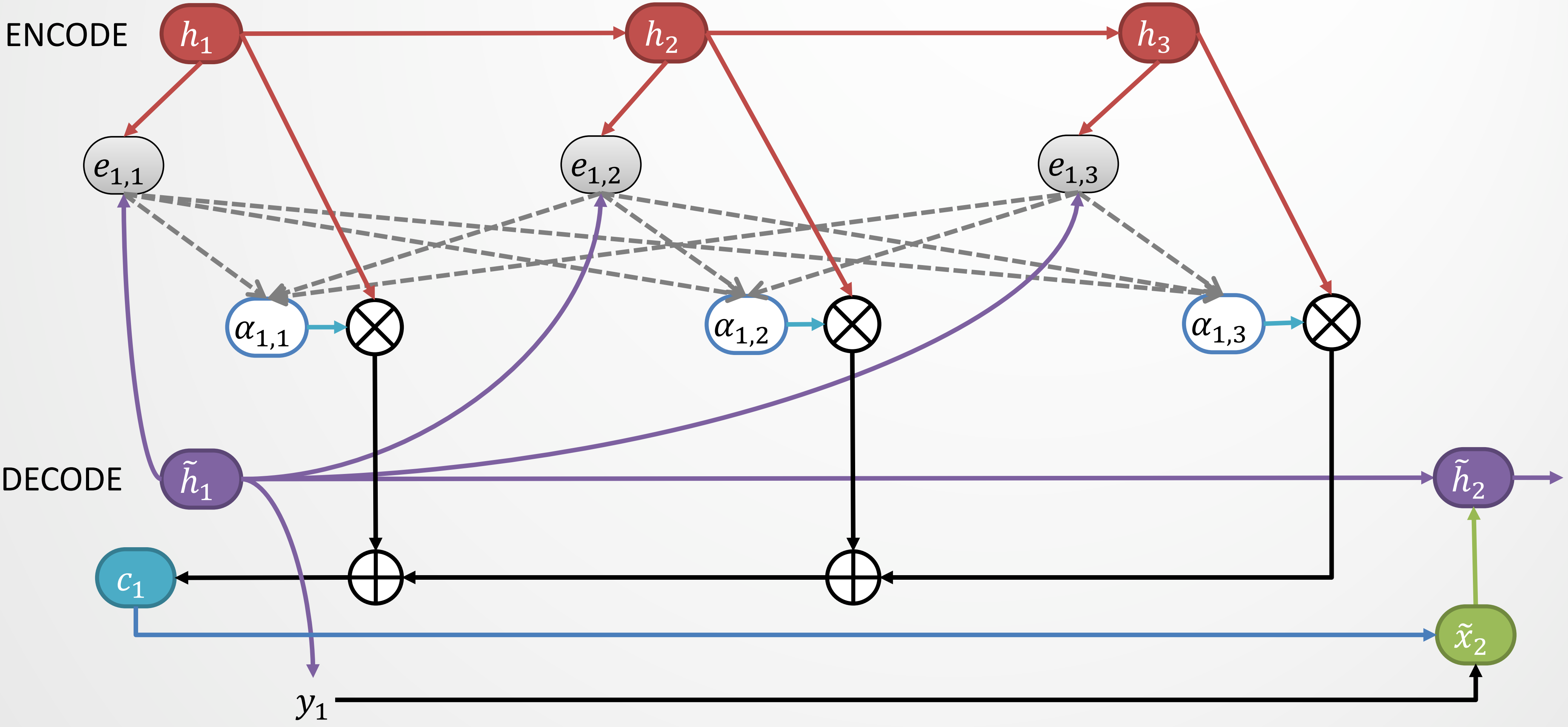


Attention mechanisms

- Input to decoder a weighted sum of **all** encoder states
- Weights determined **dynamically by decoder previous hidden state**
- $\tilde{x}_t = [T_E(y_{t-1}), c_{t-1}]$
- Context vector $c_t = \ Attend\left(\tilde{h}_t, h_{1:S}\right) = \sum_s \alpha_{t,s} h_s$
- Weights $\alpha_{t,s} = softmax\left(e_{t,1:S}, s\right) = {}^{\exp e_{t,s}}\!/_{\sum_{s'} \exp e_{t,s'}}$
- Energy scores $e_{t,s} = score\left(\tilde{h}_t, h_s\right)$
- Score function, usually $score(a, b) = |a|^{-1/2}\langle a, b\rangle$ (scaled dot-product attention)

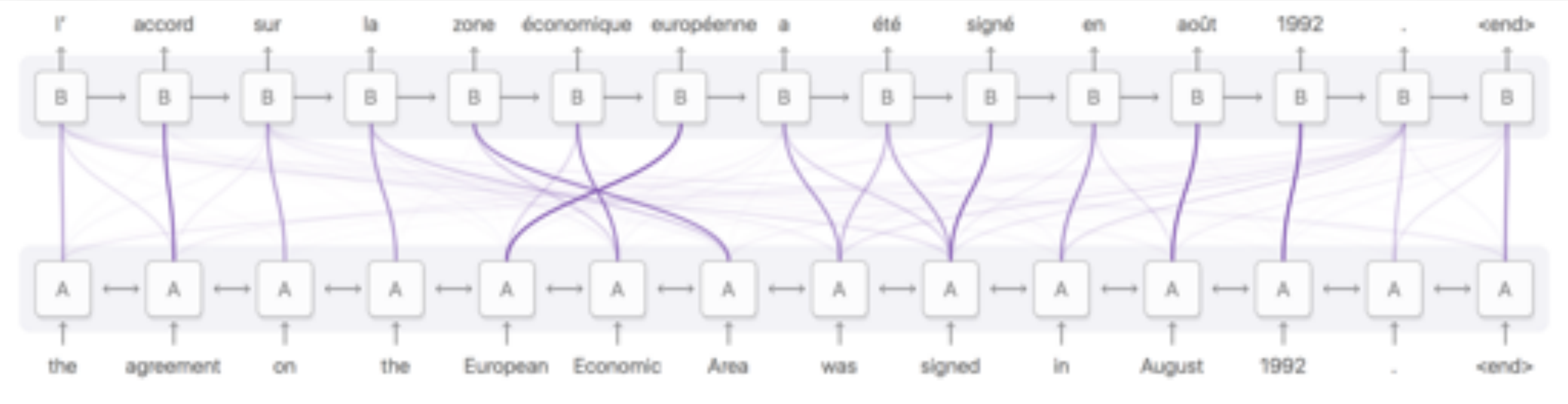
Attention example

$e_{t,s} = score(\tilde{h}_t, h_s)$ $\alpha_{t,s} = softmax(e_{t,1:S}, s)$ $c_t = \sum_s \alpha_{t,s} h_s$ $\tilde{x}_t = [T_E(y_{t-1}), c_{t-1}]$



Attention motivations

- Allow decoder to “attend” to certain areas of input when making decisions (warning: correlation ≠ causation!)
- Combines input from sequence dimension $h_{1:S}$ in a context-dependent way



Imagery from the excellent https://distill.pub/2016/augmented-rnns/#attentional-interfaces .

Multi-headed attention

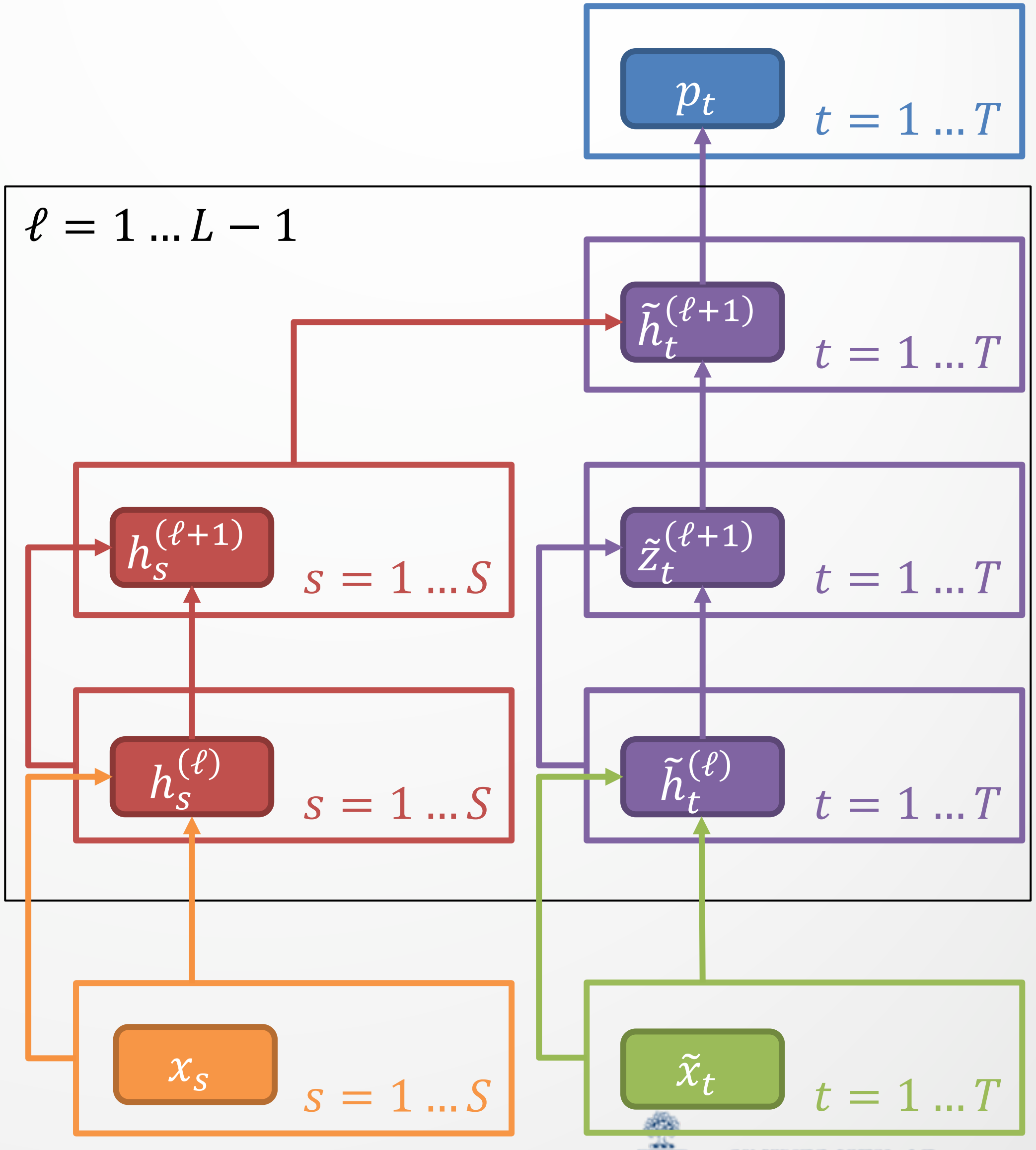
- We want to “attend to different things” for a given time step → use multi-headed attention

1. Split *N* heads $\tilde{h}_{t-1}^{(n)} = \tilde{W}^{(n)}\tilde{h}_{t-1}, h_s^{(n)} = W^{(n)}h_s$
2. Use attention: $c_{t-1}^{(n)} = Att\left(\tilde{h}_{t-1}^{(n)}, h_{1:S}^{(n)}\right)$
3. Combine for result:

$$\tilde{x}_t = \left[T_F(y_{t-1}), Qc_{t-1}^{(1:N)}\right]$$

Transformer networks

- Core idea: replace RNN with attention
- Encoder uses self-attention
  - $h_s^{(\ell+1)} \leftarrow Att_{Enc}\left(h_s^{(\ell)}, h_{1:S}^{(\ell)}\right)$
- Decoder uses self-attention, then attention with encoder
  - $\tilde{z}_t^{(\ell+1)} \leftarrow Att_{Dec1}\left(\tilde{h}_t^{(\ell)}, \tilde{h}_{1:t}^{(\ell)}\right)$
  - $\tilde{h}_t^{(\ell+1)} \leftarrow Att_{Dec2}\left(\tilde{z}_t^{(\ell+1)}, h_{1:S}^{(\ell+1)}\right)$



Transformer motivations

- RNN recurrences suffer from vanishing gradient
- Attention allows access to entire sequence
  - Better at long-term dependencies
- Lots of computation can be shared, parallelized across sequence indices
  - Feed-forward primarily + batch norm + residuals
  - See Vaswani *et al* (2017) for specific architecture

Position (in)dependence

- Attention mechanism is agnostic to sequence order
  - For permutation vector v s.t. $sorted(v) = (1,2,\ldots,V)$

$$Att(a, b_v) = Att(a, b_{1:V})$$

- **But** the order of words matters in a translation
- Solution: encode position in input

$$x_s = T_F(F_s) + \phi(s)$$

- What about decoder input?

Transformer auto-regression

- $\tilde{z}_t^{(\ell+1)} \leftarrow Att_{Dec1}\left(\tilde{h}_t^{(\ell)}, \tilde{h}_{\boxed{1:t}}^{(\ell)}\right)$
- Decoder can’t attend to future
- In teacher forcing, cannot see target directly if decoder input shifted $E_t \mapsto E_{t+1}$
- In order to decode during testing, you must
  - $y_1 \sim Decode([T_E(\text{<s>})])$
  - $y_2 \sim Decode([T_E(< s >), T_E(y_1)])$
  - Etc. until </s>

Runtime complexity

- Assume $S \approx T$

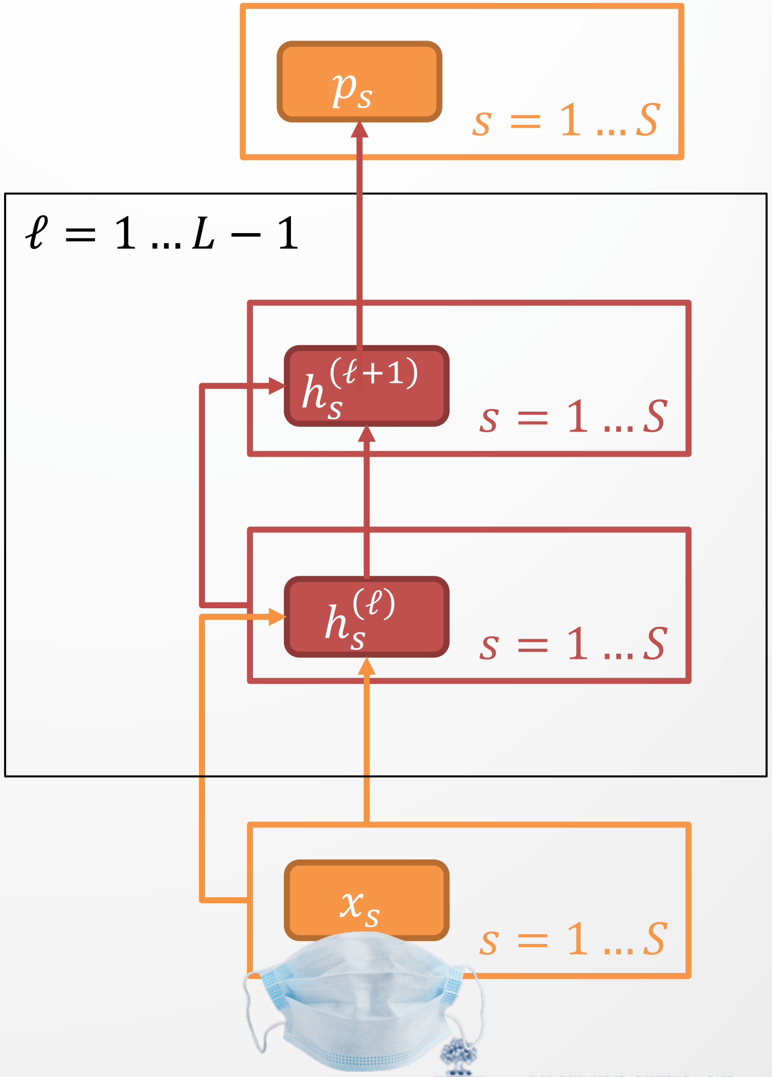
Model	Complexity	Reason
Without attention	$\boldsymbol{O}(\boldsymbol{T})$	Encoder, then decoder
With attention	$O(T^2)$	Decoder attends to all encoder states
Transformer	$O(T^2)$	Everyone attends to everyone else

- Parallelization leads to
  - Transformers quick to train, slow during decoding
  - Auto-regressive stacked RNN much slower than non-auto-regressive stacked RNNs
  - More details in CSC 421/2516

Intermezzo - BERT

(It's not an aside – it's testable!)

- **B**idirectional **E**ncoder **R**epresentations from **T**ransformers
- *Extremely* popular language representation + NLM
- Just the encoder part of the transformer model
- Learns the input that was masked

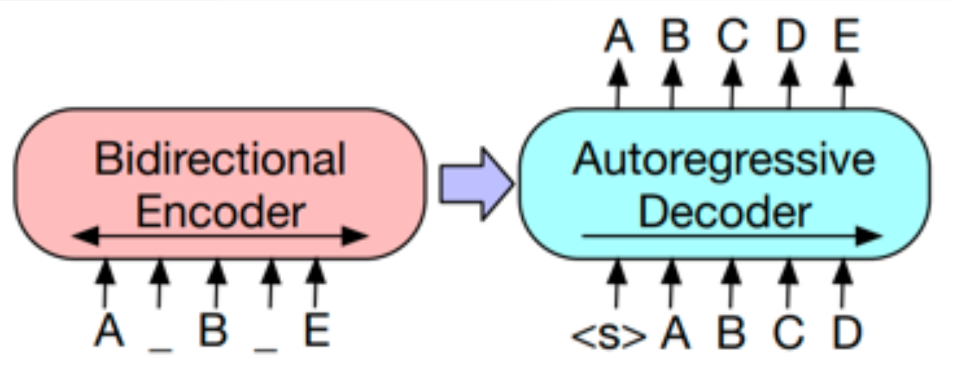


UNIVERSITY OF TORONTO

Aside – BERT → BART → NMT

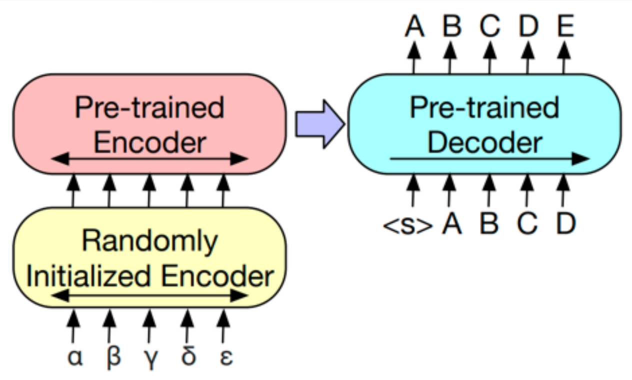
(This time it's not testable)

- Pretrained BERT language model used to re-score/fine-tune downstream NLP tasks
- Explosion of variants to BERT
- BART (Lewis *et al*, 2020) adds the decoder back to BERT, keeping the BERT objective
- Add some source language layers on top to train for NMT



BART →

BART for NMT →



Decoding in NMT

- Greedy decoding: $y_t = \text{argmax}_i(p_{t,i})$
- Can’t recover from a prior bad choice
- $\tilde{h}_t$ continuous, depends on y_{t-1}
  - Viterbi search (see HMM lecture) impossible

Beam search: top-*K* greedy

$b_{t,0}^{(k)}$: *k*-th path hidden state
$b_{t,1}^{(k)}$: *k*-th path sequence
$b_{t}^{(k\to v)}$: *k*-th path extended with token *v*

```
Given vocab $V$, decoder $\sigma$, beam width $K$
$\forall k \in [1,K].\, b_{0,0}^{(k)} \leftarrow \tilde{h}_0, \mathrm{b}_{0,1}^{(k)} \leftarrow [\text{<s>}], \log \mathrm{P}\left(b_0^{(k)}\right) \leftarrow -\mathbb{I}_{k\neq 1}\infty$
$f \leftarrow \emptyset$  # finished path indices
While $1 \notin f$:
        $\forall k \in [1,K].\, \tilde{h}_{t+1}^{(k)} \leftarrow \sigma\left(b_{t,0}^{(k)}, last\left(b_{t,1}^{(k)}\right)\right)$  # $last(x)$ gets last token in $x$
        $\forall v \in V, k \in [1,K]\backslash f.\, b_{t,0}^{(k\to v)} \leftarrow \tilde{h}_{t+1}^{(k)}, b_{t,1}^{(k\to v)} \leftarrow \left[b_{t,1}^{(k)}, v\right]$
                $\log P\left(b_t^{(k\to v)}\right) \leftarrow \log P(y_{t+1} = v|\tilde{h}_{t+1}^{(k)}) + \log P\left(b_t^{(k)}\right)$
        $\forall v \in V, k \in f.\, b_t^{(k\to v)} \leftarrow b_t^{(k)}, \log P\left(b_t^{(k\to v)}\right) \leftarrow \log P\left(b_t^{(k)}\right) - \mathbb{I}_{v\neq \text{</s>}}\infty$
        $\forall k \in [1,K].\, b_{t+1}^{(k)} \leftarrow \operatorname{argmax}^{k}_{b_t^{(k'\to v)}} \log P\left(b_t^{(k'\to v)}\right)$  # $k$-th max $b_t^{(k'\to v)}$
        $f \leftarrow \{k \in [1,K] | last\left(b_{t+1}^{(k)}\right) = \text{</s>}\}$
        $t \leftarrow t+1$
Return $b_{t,1}^{(1)}$
```

*Other completion criteria exist (e.g. $t \leq T$, finish some # of paths)

Beam search example (*t=1*)

$V = \{\mathrm{H}, \mathrm{A}, </\mathrm{s}>\}$, K=2

k	$b_{0,1}^{(k)}$	$P\left(b_0^{(k)}\right)$
1	[<s>]	1
2	[<s>]	0

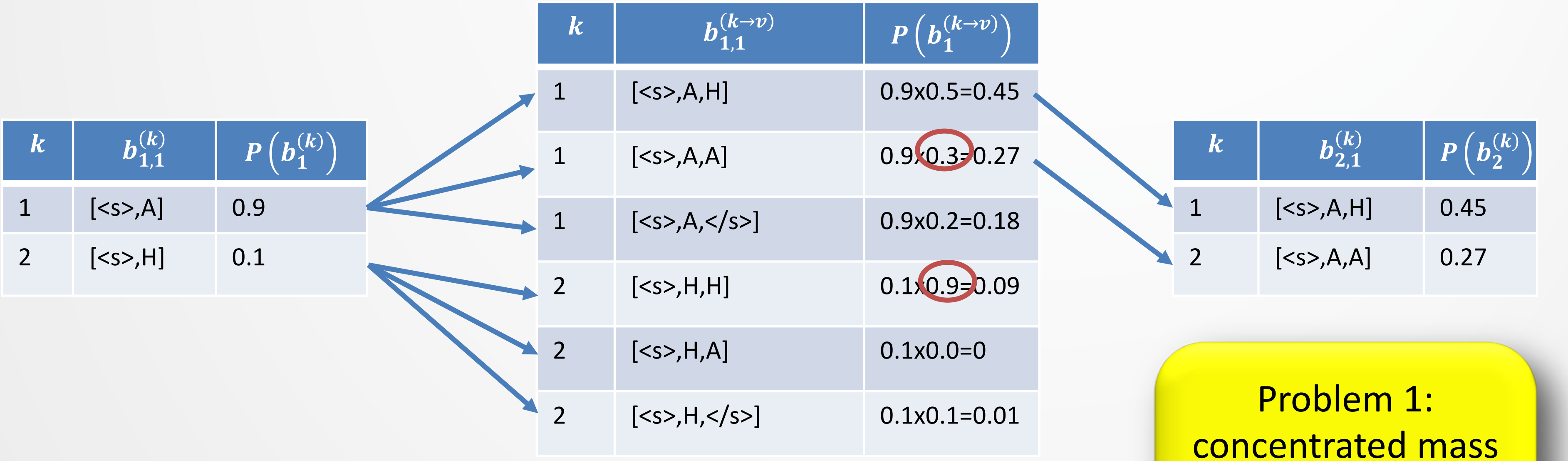
k	$b_{0,1}^{(k\to v)}$	$P\left(b_0^{(k\to v)}\right)$
1*	[<s>,H]	1x0.1=0.1
1*	[<s>,A]	1x0.9=0.9
1*	[<s>,</s>]	1x0=0
2	[<s>,H]	0x0.1=0
2	[<s>,A]	0x0.9=0
2	[<s>,</s>]	0x0=0

k	$b_{1,1}^{(k)}$	$P\left(b_1^{(k)}\right)$
1	[<s>,A]	0.9
2	[<s>,H]	0.1

*Note $\forall k. \sum_v P\left(b_t^{(k\to v)}\right) = 1$

Beam search example (*t*=2)

$V = \{\mathrm{H}, \mathrm{A}, </\mathrm{s}>\}$, K=2



k	$b_{1,1}^{(k)}$	$P\left(b_1^{(k)}\right)$
1	[<s>,A]	0.9
2	[<s>,H]	0.1

k	$b_{1,1}^{(k\to v)}$	$P\left(b_1^{(k\to v)}\right)$
1	[<s>,A,H]	0.9x0.5=0.45
1	[<s>,A,A]	0.9x0.3=0.27
1	[<s>,A,</s>]	0.9x0.2=0.18
2	[<s>,H,H]	0.1x0.9=0.09
2	[<s>,H,A]	0.1x0.0=0
2	[<s>,H,</s>]	0.1x0.1=0.01

k	$b_{2,1}^{(k)}$	$P\left(b_2^{(k)}\right)$
1	[<s>,A,H]	0.45
2	[<s>,A,A]	0.27

Problem 1: concentrated mass on a prefix creates near identical hypotheses

Beam search example (*t=3*)

$V = \{\mathrm{H}, \mathrm{A}, </\mathrm{s}>\}$, K=2

k	$b_{2,1}^{(k)}$	$P\left(b_2^{(k)}\right)$
1	[<s>,A,H]	0.45
2	[<s>,A,A]	0.27

k	$b_{2,1}^{(k\to v)}$	$P\left(b_2^{(k\to v)}\right)$
1	[<s>,A,H,H]	0.45x0.5=0.225
1	[<s>,A,H,A]	0.45x0.3=0.135
1	[<s>,A,H,</s>]	0.45x0.2=0.09
2	[<s>,A,A,H]	0.27x0.2=0.054
2	[<s>,A,A,A]	0.27x0.2=0.054
2	[<s>,A,A,</s>]	0.27x0.6=0.162

k	$b_{3,1}^{(k)}$	$P\left(b_3^{(k)}\right)$
1	[<s>,A,H,H]	0.225
2	[<s>,A,A,</s>]	0.162

Beam search example (*t=4*)

$V = \{\mathrm{H}, \mathrm{A}, </\mathrm{s}>\}$, K=2

k	$b_{3,1}^{(k)}$	$P\left(b_3^{(k)}\right)$
1	[<s>,A,H,H]	0.225
2	[<s>,A,A,</s>]	0.162

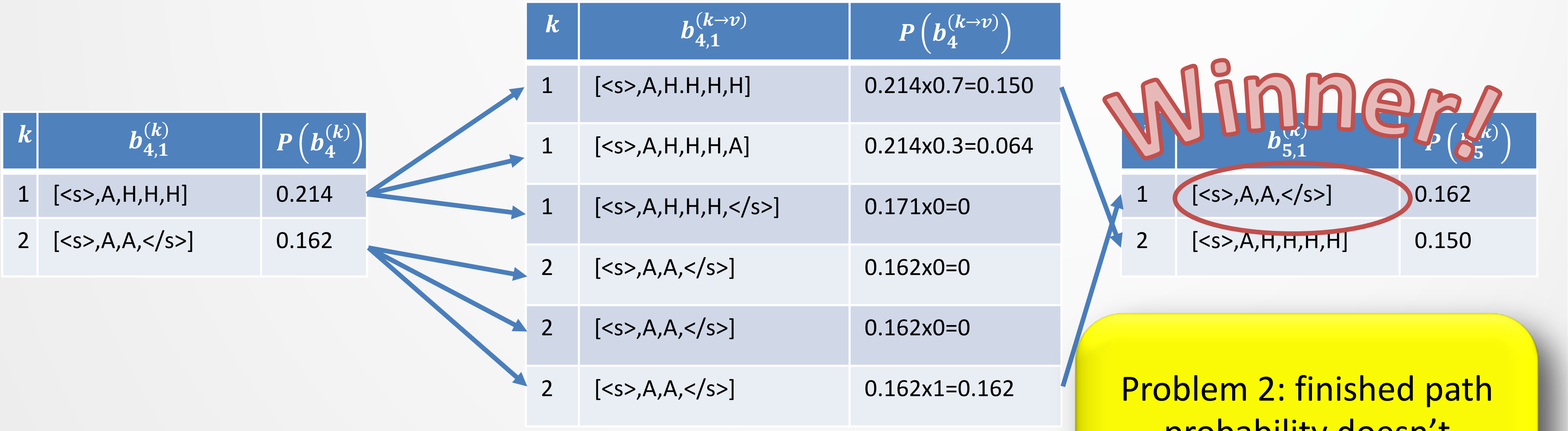
k	$b_{3,1}^{(k\to v)}$	$P\left(b_3^{(k\to v)}\right)$
1	[<s>,A,H,H,H]	0.225x0.9=0.214
1	[<s>,A,H,H,A]	0.225x0.05=0.01
1	[<s>,A,H,H,</s>]	0.18x0=0
2*	[<s>,A,A,</s>]	0.162x0=0
2*	[<s>,A,A,</s>]	0.162x0=0
2*	[<s>,A,A,</s>]	0.162x1=0.162

k	$b_{4,1}^{(k)}$	$P\left(b_4^{(k)}\right)$
1	[<s>,A,H,H,H]	0.214
2	[<s>,A,A,</s>]	0.162

*Since k=2 is finished

Beam search example (*t*=5)

$V = \{\mathrm{H}, \mathrm{A}, </\mathrm{s}>\}$, K=2



k	$b_{4,1}^{(k)}$	$P\left(b_4^{(k)}\right)$
1	[<s>,A,H,H,H]	0.214
2	[<s>,A,A,</s>]	0.162

k	$b_{4,1}^{(k\to v)}$	$P\left(b_4^{(k\to v)}\right)$
1	[<s>,A,H.H,H,H]	0.214x0.7=0.150
1	[<s>,A,H,H,H,A]	0.214x0.3=0.064
1	[<s>,A,H,H,H,</s>]	0.171x0=0
2	[<s>,A,A,</s>]	0.162x0=0
2	[<s>,A,A,</s>]	0.162x0=0
2	[<s>,A,A,</s>]	0.162x1=0.162

k	$b_{5,1}^{(k)}$	$P\left(b_5^{(k)}\right)$
1	[<s>,A,A,</s>]	0.162
2	[<s>,A,H,H,H,H]	0.150

Problem 2: finished path probability doesn't decrease → preference for shorter paths

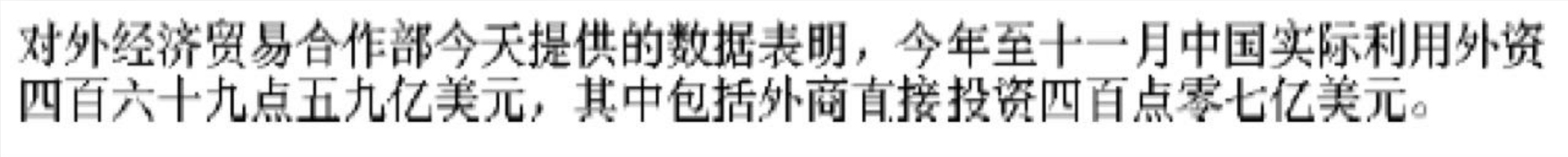
Sub-words

- Out-of-vocabulary words can be handled by breaking up words into parts
  - “abwasser+behandlungs+anlange” → “water sewage plant”
- Sub-word units are built out of combining characters (like phrases!)
- Popular approaches include
  - Byte Pair Encoding: “Neural machine translation of rare words with subword units,” 2016. Sennrich *et al.*
  - Wordpieces: “Google’s neural machine translation system: bridging the gap between human and machine translation,” 2016. Wu *et al.*

Aside – advanced NMT

- Modifications to beam search
  - “Diverse beam search,” 2018. Vijayakumar *et al.*
- Exposure bias
  - “Optimal completion distillation,” 2018. Sabour *et al.*
- Back translation
  - “Improving neural machine translation models with monolingual data,” 2016. Senrich *et al*.
- “Non-autoregressive neural machine translation,” 2018. Gu *et al*.
- “Unsupervised neural machine translation,” 2018. Artetxe *et al*.
- “BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension,” 2020. Lewis *et al.*

Evaluation of MT systems



对外经济贸易合作部今天提供的数据表明，今年至十一月中国实际利用外资四百六十九点五九亿美元，其中包括外商直接投资四百点零七亿美元。

Human	According to the data provided today by the Ministry of Foreign Trade and Economic Cooperation, as of November this year, China has actually utilized 46.959B US dollars of foreign capital, including 40.007B US dollars of direct investment from foreign businessmen.
IBM4	The Ministry of Foreign Trade and Economic Cooperation, including foreign direct investment 40.007B US dollars today provide data include that year to November China actually using foreign 46.959B US dollars and
Yamada/ Knight	Today's available data of the Ministry of Foreign Trade and Economic Cooperation shows that China's actual utilization of November this year will include 40.007B US dollars for the foreign direct investment among 46.959B US dollars in foreign capital.

How can we objectively compare the quality of two translations?

Automatic evaluation

- We want an **automatic** and effective method to **objectively** rank competing translations.
  - **Word Error Rate (WER)** measures the number of erroneous word **insertions**, **deletions**, **substitutions** in a translation.
    - E.g., **Reference**: *how to recognize speech*
      Translation: *how understand a speech*
    - **Problem**: There are many possible valid translations.
      (There's no need for an exact match)

Challenges of evaluation

- **Human judges**: expensive, slow, non-reproducible (different judges – different biases).

- Multiple valid translations, e.g.:
  - **Source**: *Il s’agit d’un guide qui assure que l’armée sera toujours fidèle au Parti*
  - **T1**: *It is a guide to action that ensures that the military will forever heed Party commands*
  - **T2**: *It is the guiding principle which guarantees the military forces always being under command of the Party*

BLEU evaluation

- **BLEU (BiLingual Evaluation Understudy)** is an automatic and popular method for evaluating MT.
  - It uses **multiple** human **reference** translations, and looks for local matches, allowing for phrase movement.

  - **Candidate:** *n.* a translation produced by a machine.

- There are a few parts to a **BLEU score**…

Example of BLEU evaluation

- **Reference 1**: *It is a guide to action that ensures that the military will forever heed Party commands*
- **Reference 2**: *It is the guiding principle which guarantees the military forces always being under command of the Party*
- **Reference 3**: *It is the practical guide for the army always to heed the directions of the party*

- **Candidate 1**: *It is a guide to action which ensures that the military always obeys the commands of the party*
- **Candidate 2**: *It is to insure the troops forever hearing the activity guidebook that party direct*

BLEU: Unigram precision

- The **unigram precision** of a candidate is

$$\frac{C}{N}$$

where N is the number of words in the **candidate**
and C is the number of words in the **candidate** which are in **at least one reference**.

- e.g., **Candidate 1**: *It is a guide to action which ensures that the military always obeys the commands of the party*
  - **Unigram precision** $= \frac{17}{18}$

    (*obeys* appears in none of the three references).

BLEU: Modified unigram precision

- **Reference 1**: *The lunatic is on the grass*
- **Reference 2**: *There is a lunatic upon the grass*
- **Candidate**: *The the the the the the the*
  - Unigram precision $= \frac{7}{7} = 1$ 🙁

- **Capped unigram precision:**

  A candidate word type w can only be correct a **maximum** of $cap(w)$ times.

  - e.g., with $\boldsymbol{cap(the) = 2}$, the above gives

    $$p_1 = \frac{2}{7}$$

BLEU: Generalizing to *N*-grams

- Generalizes to higher-order *N*-grams.
  - **Reference 1**: ***It is*** *a guide to action that ensures that the military will forever heed Party commands*
  - **Reference 2**: ***It is*** *the guiding principle which guarantees the military forces always being under command of the Party*
  - **Reference 3**: ***It is*** *the practical guide for the army always to heed the directions of the party*

Bigram precision, p_2

- **Candidate 1**: ***It is*** *a guide to action which ensures that the military always obeys the commands of the party*
- **Candidate 2**: ***It is*** *to insure the troops forever hearing the activity guidebook that party direct*

$p_2 = 10/17$

$p_2 = 1/13$

BLEU: Precision is not enough

- **Reference 1**: *It is a guide to action that ensures that the military will forever heed Party commands*
- **Reference 2**: *It is the guiding principle which guarantees the military forces always being under command **of the** Party*
- **Reference 3**: *It is the practical guide for the army always to heed the directions **of the** party*

- **Candidate 1**: ***of the***

Unigram precision, $p_1 = \frac{2}{2} = 1$ Bigram precision, $p_2 = \frac{1}{1} = 1$

BLEU: Brevity

- Solution: Penalize brevity.
- **Step 1**: for each candidate,
  find the reference **most similar in length**.
- **Step 2**: $\boldsymbol{c_i}$ is the length of the i^{th} candidate, and
  $\boldsymbol{r_i}$ is the nearest length among the references,

$$brevity_i = \frac{r_i}{c_i}$$

Bigger = too brief

- **Step 3**: multiply precision by the (0..1) **brevity penalty**:

$$BP_i = \begin{cases} 1 & \text{if } brevity_i < 1 \\ e^{1-brevity_i} & \text{if } brevity_i \geq 1 \end{cases}$$

$(r_i < c_i)$

$(r_i \geq c_i)$

BLEU: Final score

- On slide 67, $r_1 = 16, r_2 = 17, r_3 = 16$, and $c_1 = 18$ and $c_2 = 14$,

$$brevity_1 = \frac{17}{18} \qquad BP_1 = 1$$

$$brevity_2 = \frac{16}{14} \qquad BP_2 = e^{1-\left(\frac{8}{7}\right)} = 0.8669$$

- **Final score** of candidate C:

$$\boxed{BLEU_C = BP_C \times (p_1 p_2 \ldots p_n)^{1/n}}$$

where p_n is the n-gram precision. (You can set n empirically)

Example: Final BLEU score

- **Reference 1:** *I am afraid Dave*
  Reference 2: *I am scared Dave*
  Reference 3: *I have fear David*
  Candidate: *I fear David*

Assume $cap(\cdot) =$ 2 for all *N*-grams

- $brevity = \frac{4}{3} \geq 1$ so $BP = e^{1-\left(\frac{4}{3}\right)}$

Also assume BLEU order $n = 2$

- $p_1 = \frac{1+1+1}{3} = 1$
- $p_2 = \frac{1}{2}$
- $BLEU = BP(p_1 p_2)^{\frac{1}{2}} = e^{1-\left(\frac{4}{3}\right)} \left(\frac{1}{2}\right)^{\frac{1}{2}} \approx 0.5067$

Aside – Corpus-level BLEU

- To calculate BLEU over M source sentences (assuming one candidate per source)…
- $BLEU \neq \frac{1}{M}\sum_{m=1}^{M} BLEU_m$
- Sum statistics over *all* sources
  - m indexes m-th source sentence, drop candidate index i
  - $p_n = \frac{\sum_{m=1}^{M} capped\_true\_ngram\_count_m}{\sum_{m=1}^{M} N_m}$
  - $r = \sum_{m=1}^{M} r_m$
  - $c = \sum_{m=1}^{M} c_m$
  - $brevity = r/c$
- **We won't ask you to calculate it this way**

BLEU: summary

- BLEU is a geometric mean over n-gram precisions.
  - These precisions are **capped** to avoid strange cases.
    - E.g., the translation “*the the the the*” is not favoured.
  - This geometric mean is **weighted** so as not to favour unrealistically short translations, e.g., “*the*”
- Initially, evaluations showed that BLEU predicted human judgements very well, but:
  - People started **optimizing** MT systems to **maximize** BLEU. Correlations between BLEU and humans **decreased**.