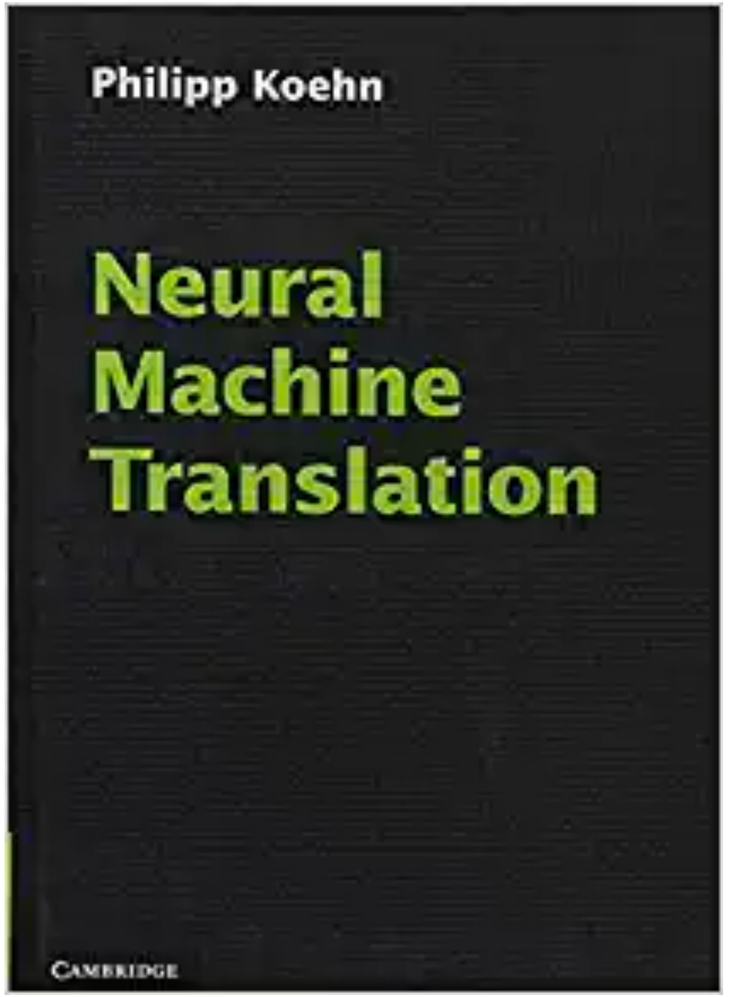
Encoder-Decoder (aka Seq2Seq) and Attention

Wei Xu

(many slides from Greg Durrett)

This Lecture

- Machine Translation
- Sequence-to-Sequence Model
- Attention Mechanism



- Reading — Eisenstein 18.3-18.5

MT Basics

MT Basics

People's Daily, August 30, 2017

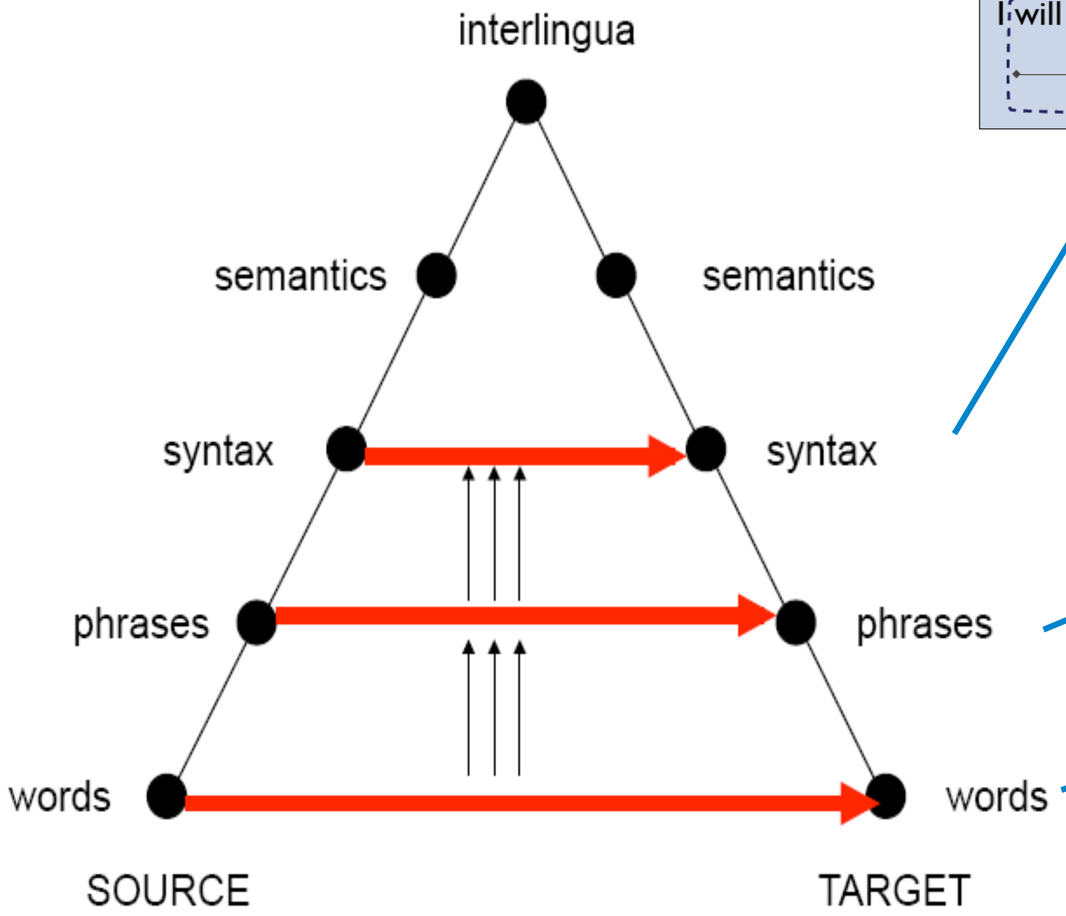
Trump Pope family watch a hundred years a year in the White House balcony

MT Basics

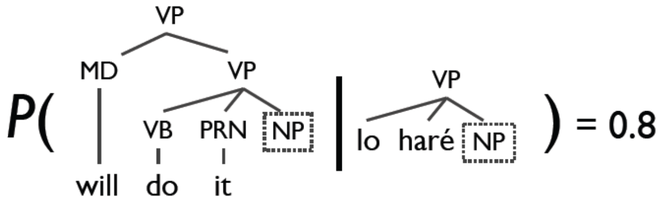
People's Daily, August 30, 2017

Trump and his family watch the once-in-a-century total solar eclipse from the White House balcony

Levels of Transfer: Vauquois Triangle (1968)



Yo lo haré mañana
I will do it tomorrow
NP
VP



Yo lo haré mañana
I will do it tomorrow

English (E)	*P*(E | lo haré)
will do it	0.8
will do so	0.2

Yo lo haré mañana
I will do it tomorrow

English (E)	*P*(E | mañana)
tomorrow	0.7
morning	0.3

Slide credit: Dan Klein

MT Ideally

- I have a friend => $\exists x$ `friend(x,self)` => J'ai un ami
  J'ai une amie
  - May need information you didn't think about in your representation
  - Hard for semantic representations to cover everything
- Everyone has a friend => $\exists x \forall y$ `friend(x,y)` => Tout le
  $\forall x \exists y$ `friend(x,y)` monde a un ami
  - Can often get away without doing all disambiguation — same ambiguities may exist in both languages

The Birth of MT

- Decipherment of the German ENIGMA code by the British team that includes Alan Turing

- "It is very tempting to say that a book written in Chinese is simply a book written in English which was coded into the 'Chinese Code'. If we have useful methods for solving almost any cryptographic problem, may it not be that with proper interpretation we already have useful methods for translation?”
  Weaver (1955)

Daniel Stein (2013)

History of MT

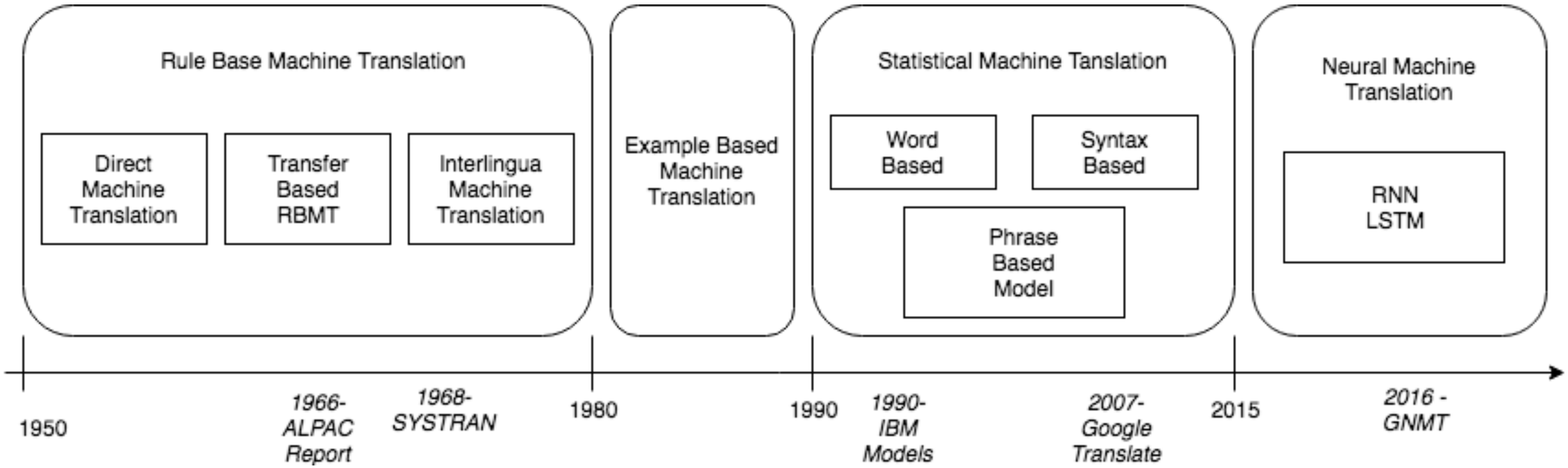


Image Credit: https://towardsdatascience.com/evolution-of-machine-translation-5524f1c88b25

History of MT

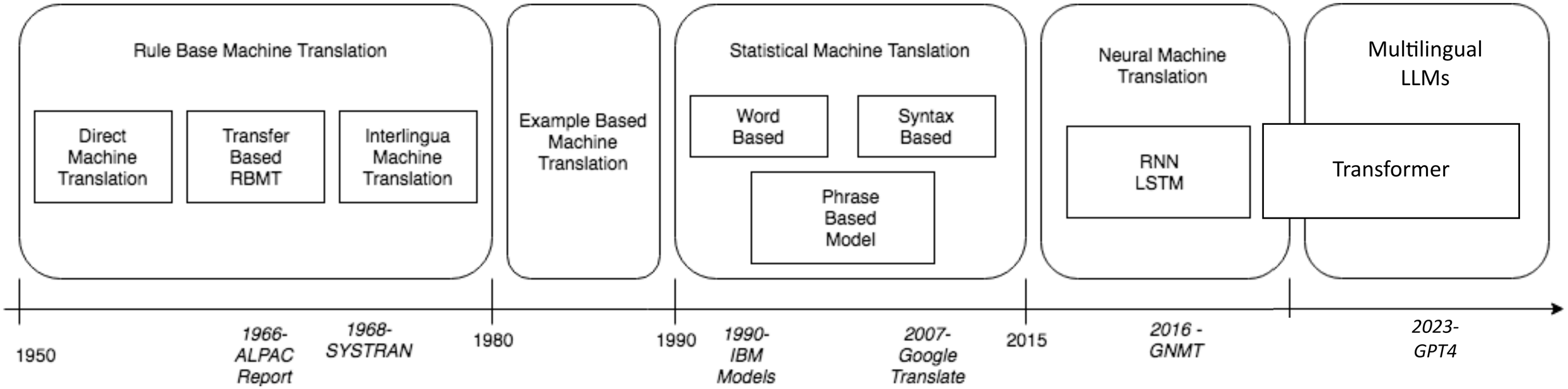
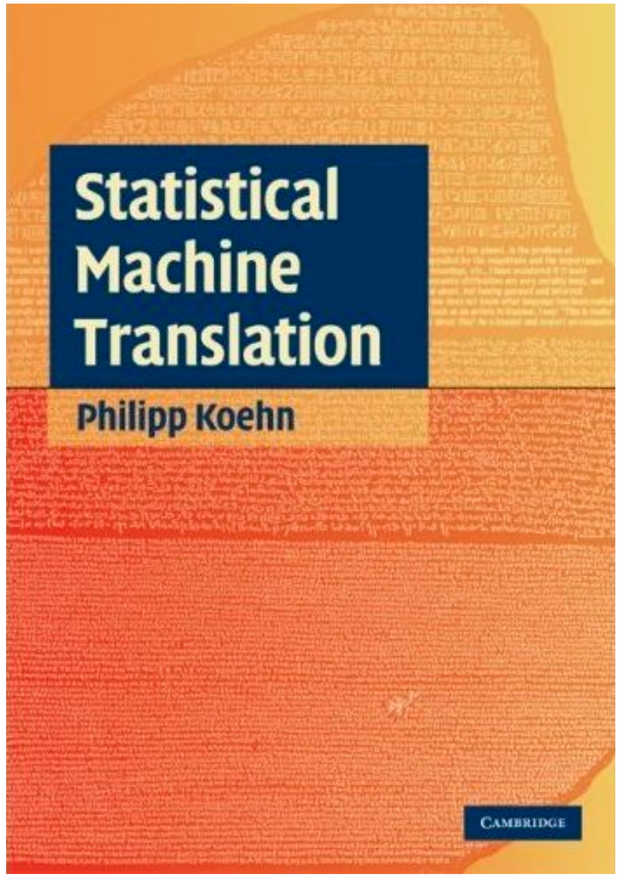


Image Credit: https://towardsdatascience.com/evolution-of-machine-translation-5524f1c88b25

Parallel Training Corpus

		facing with the swelling flow of through traffic zooming past their doors .		retahila de inconvenientes que más y más gente tiene que soportar por el tráfico que pasa por delante_de sus casas , que aumenta a_diario .
5	#77501757	Weekend traffic bans and traffic **jams** are a curse to road transport .	#74765580	Las prohibiciones de conducir los fines de semana y los embotellamientos asolan el transporte por carretera .
6	#79500725	Some people also want to recoup the cost of traffic **jams** from those who get stuck in them , according to the ' polluter pays ' principle .	#76764676	Algunos son partidarios de que incluso los costes ocasionados por los atascos se carguen a el ciudadano que se encuentra atrapado en ellos , de conformidad con el principio de que " quien contamina paga " .
7	#79500765	I think this is an excellent principle and I would like to see it applied in full , but not to traffic **jams** .	#76764713	Me parece un principio acertado y estoy dispuesta a aplicarlo íntegramente , pero no sobre los atascos , ya_que éstos son un claro indicio de el fracaso de la política gubernamental en_materia_de infraestructuras .
8	#79500768	Traffic **jams** are indicative of failed government policy on the infrastructure front , which is why the government itself , certainly in the Netherlands , must be regarded as the polluter .	#76764747	Por eso es preciso subrayar que en estos casos quien contamina es el propio Gobierno , a el menos en los Países_Bajos .
9	#81309716	This would increase traffic **jams** , weaken road safety and increase costs .	#78586130	Esto aumentaría los atascos , mermaría la seguridad vial e incrementaría los costes .
10	#81997391	In the previous legislature , Parliament gave its opinion on the Commission ' s proposals on the simplification of vertical directives on sugar , honey , fruit juices , milk and **jams** .	#79281114	En efecto , durante la precedente legislatura , el Parlamento se manifestó sobre las propuestas de la Comisión relativas a la simplificación de directivas verticales sobre el azúcar , la miel , los zumos de frutas , la leche y las confituras .
11	#81998167	For **jams** , I personally reintroduced an amendment that was not accepted by the Committee on the Environment , Public Health and Consumer Policy , but which I hold to .	#79281936	Para las confituras , yo personalmente volví a introducir una enmienda que no fue aceptada por la Comisión_de_Medio_Ambiente , Salud_Pública y Política_de_el_Consumidor , pero que es importante para mí .
12	#81998209	It concerns not accepting the general use of a chemical flavouring in **jams** and marmalades , that is vanillin .	#79281966	Se trata de no aceptar la utilización generalizada de un aroma químico en las confituras y " marmalades " , a saber , la vainillina .
13	#82800065	This is highlighted particularly in towns where it is necessary to find ways of solving environmental problems and the difficulties caused by traffic **jams** .	#80085988	Esto se pone_de_relieve aún más en las ciudades , en las que hay que encontrar medios para eliminar los inconvenientes derivados de los problemas medioambientales y de la congestión de el tráfico .

Phrase-based MT



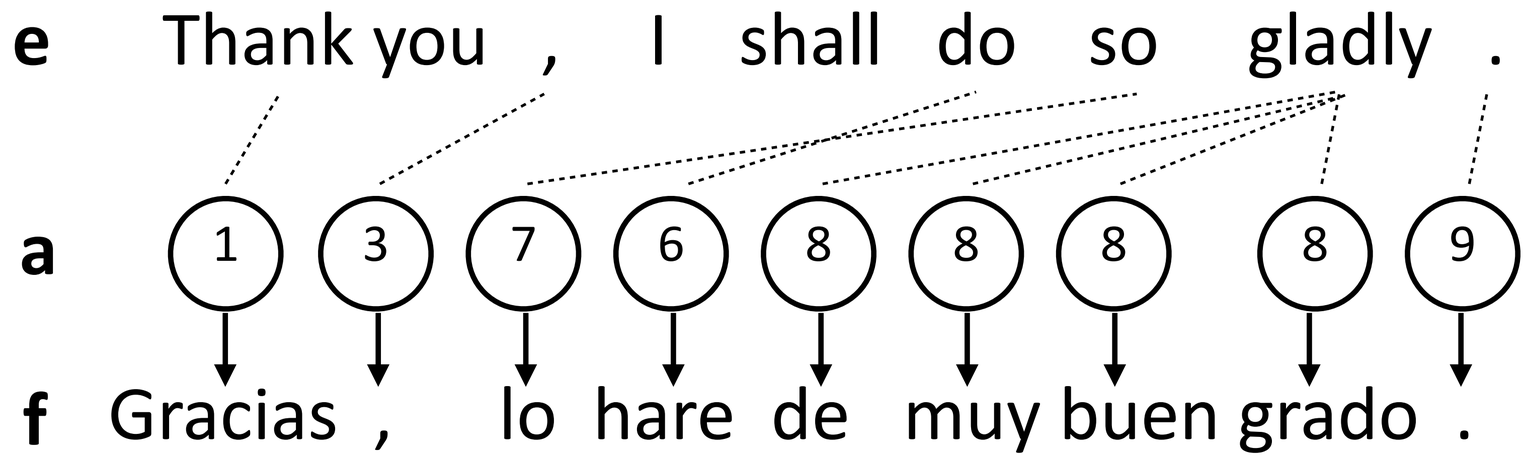
Phrase-Based MT

- Key idea: translation words better the bigger chunks you use
- Remember phrases from training data, translate piece-by-piece and stitch those pieces together to translate
  - How to identify phrases? Word alignment over source-target bitext
  - How to stitch together? Language model over target language
  - Decoder takes phrases and a language model and searches over possible translations
- NOT like standard discriminative models (take a bunch of translation pairs, learn a ton of parameters in an end-to-end way)

Word Alignment: IBM Model 1

- Each “Foreign” word is aligned to *at most* one English word

$$P(\mathbf{f}, \mathbf{a}|\mathbf{e}) = \prod_{i=1}^{n} P(f_i|e_{a_i})P(a_i)$$



- Set P(a) uniformly (no prior over good alignments) = 1 / (#words in e + 1)
- $P(f_i|e_{a_i})$: word translation probability. Learn with EM (Eisenstein ch 18.2.2) Brown et al. (1993)

Word Alignment

- Find contiguous sets of aligned words in the two languages that don't have alignments to other words

de assister à la runion et ||| to attend the meeting and

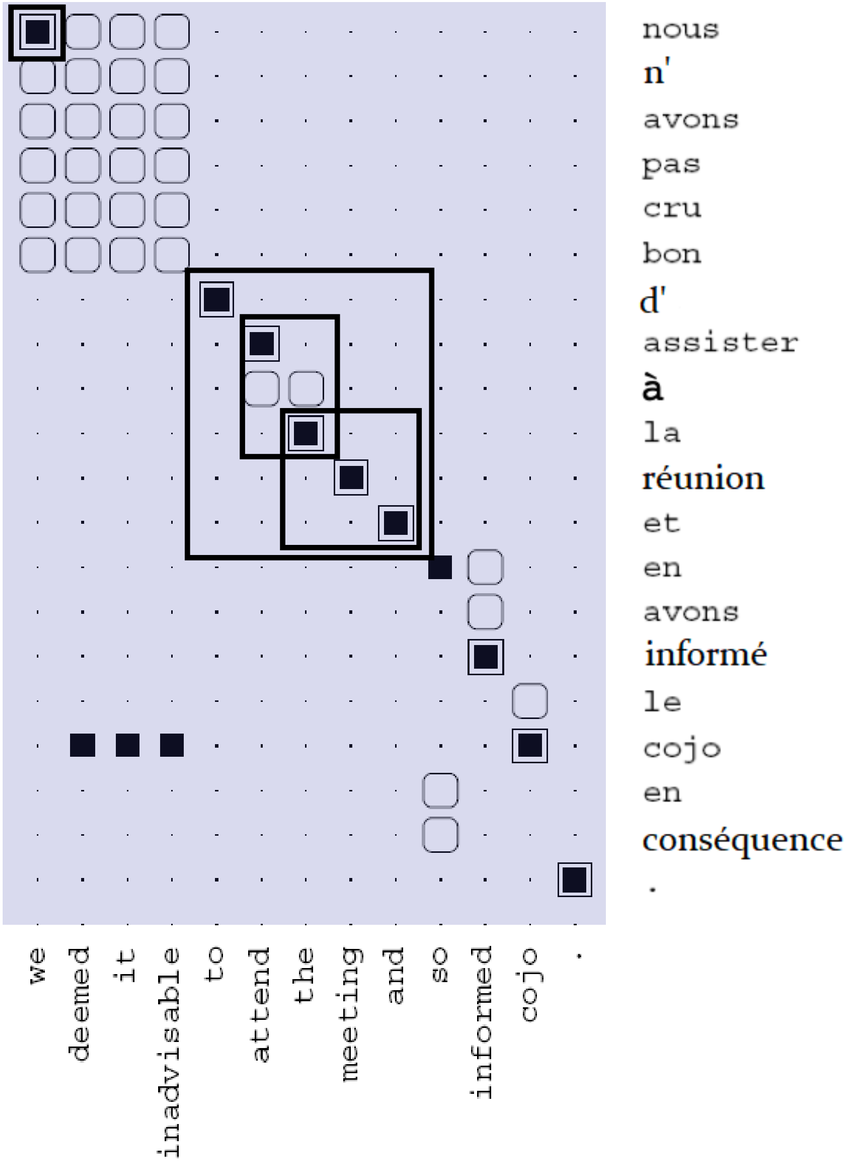
assister à la runion ||| attend the meeting

la runion and ||| the meeting and

nous ||| we

…

- Lots of phrases possible, count across all sentences and score by frequency

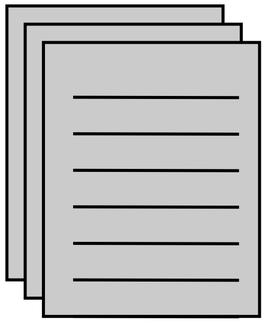


Phrase-Based MT

- Goal: translate from Foreign language to English

cat ||| chat ||| 0.9
the cat ||| le chat ||| 0.8
dog ||| chien ||| 0.8
house ||| maison ||| 0.6
my house ||| ma maison ||| 0.9
language ||| langue ||| 0.9
…

Phrase table $P(f|e)$



Unlabeled English data

Language model $P(e)$

$$P(e|f) \propto P(f|e)P(e)$$

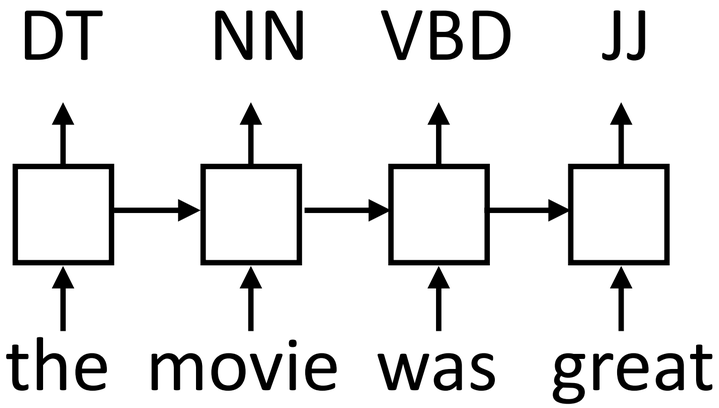
Noisy channel model: combine scores from translation model + language model to translate foreign to English

"Translate faithfully but make fluent English"

Seq2Seq Models

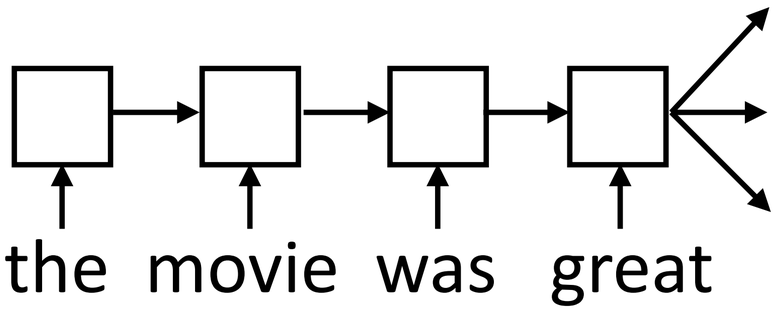
Recap: RNN

- Transducer: make some prediction for each element in a sequence



output **y** = score for each tag, then softmax

- Encoder: encode a sequence into a fixed-sized vector and use that for some purpose



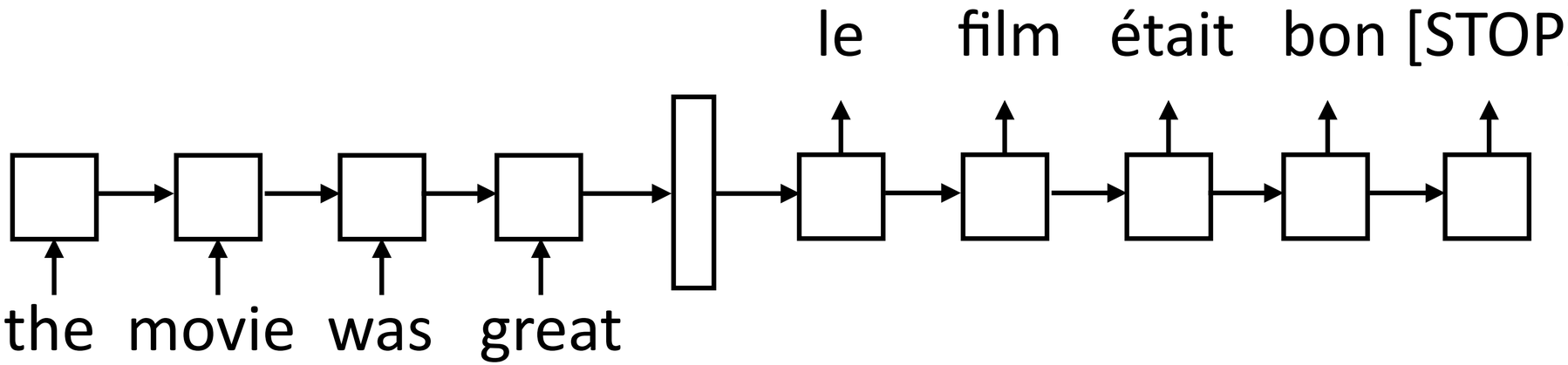
predict sentiment (matmul + softmax)

translate

paraphrase/compress

Encoder-Decoder

- Encode a sequence into a fixed-sized vector

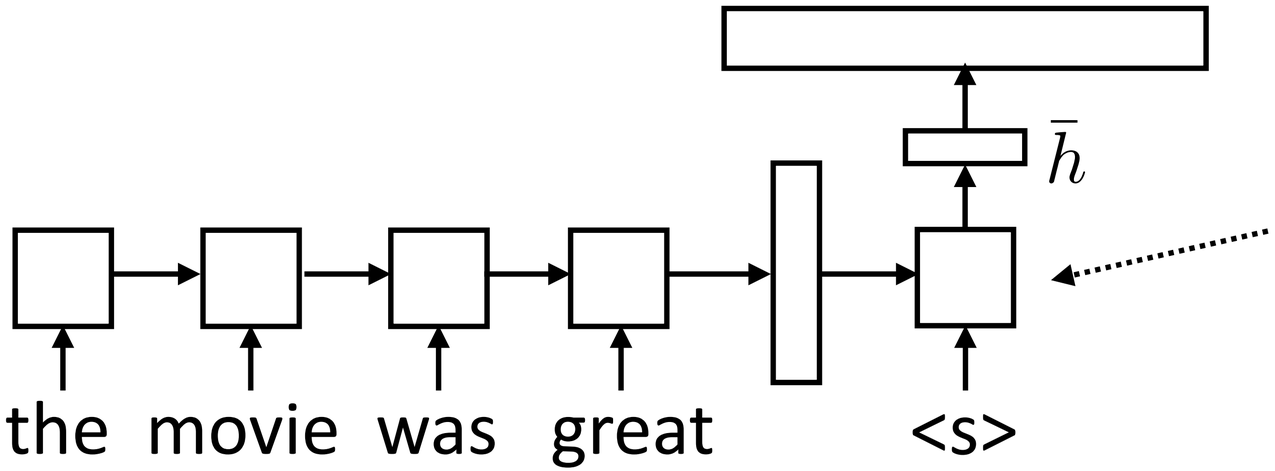


- Now use that vector to produce a series of tokens as output from a separate LSTM *decoder*
- Machine translation, NLG, summarization, dialog, and many other tasks (e.g., semantic parsing, syntactic parsing) can be done using this framework.

Sutskever et al. (2014)

Model

- Generate next word conditioned on previous word as well as hidden state
- W size is |vocab| x |hidden state|, softmax over entire vocabulary



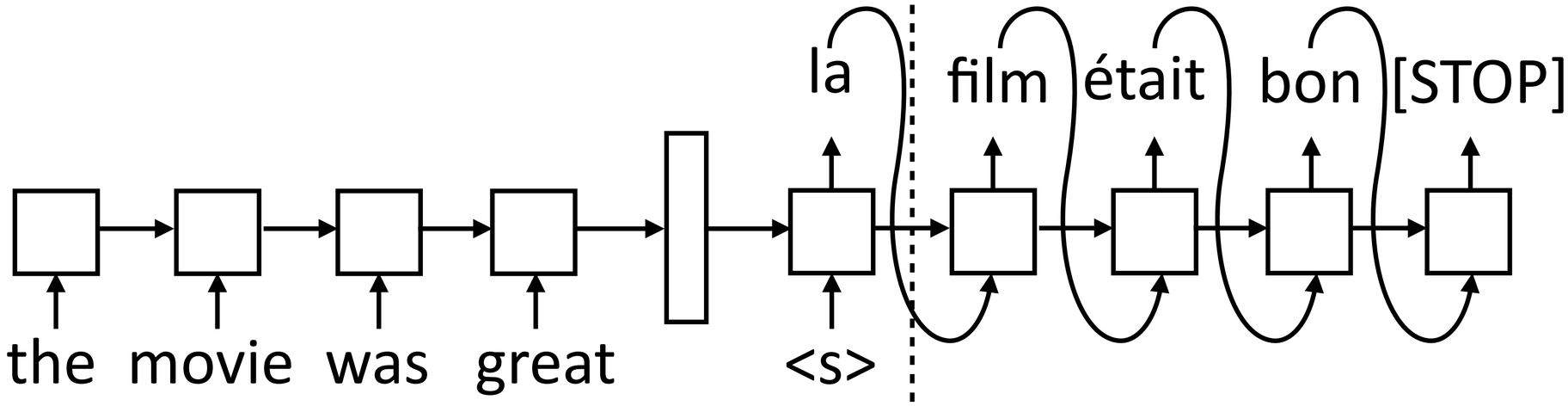
$$P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) = \text{softmax}(W\bar{h})$$

$$P(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{n} P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1})$$

Decoder has separate parameters from encoder, so this can learn to be a language model (produce a plausible next word given current one)

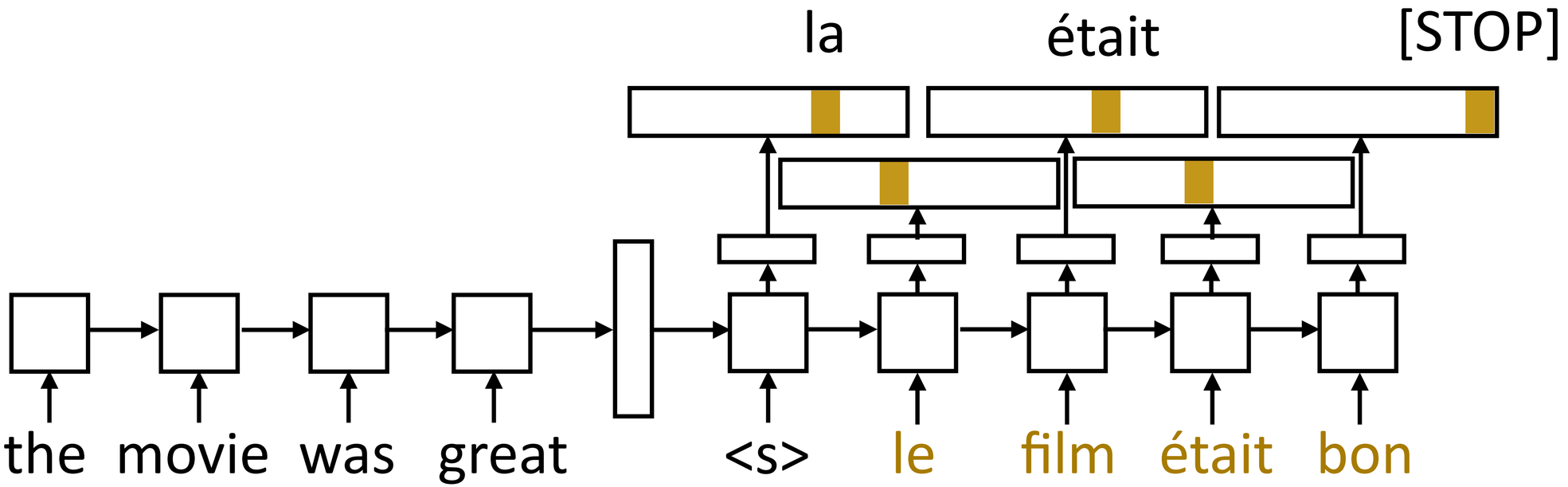
Inference

- Generate next word conditioned on previous word as well as hidden state



- During inference: need to compute the argmax over the word predictions and then feed that to the next RNN state
- Decoder is advanced one state at a time until [STOP] is reached

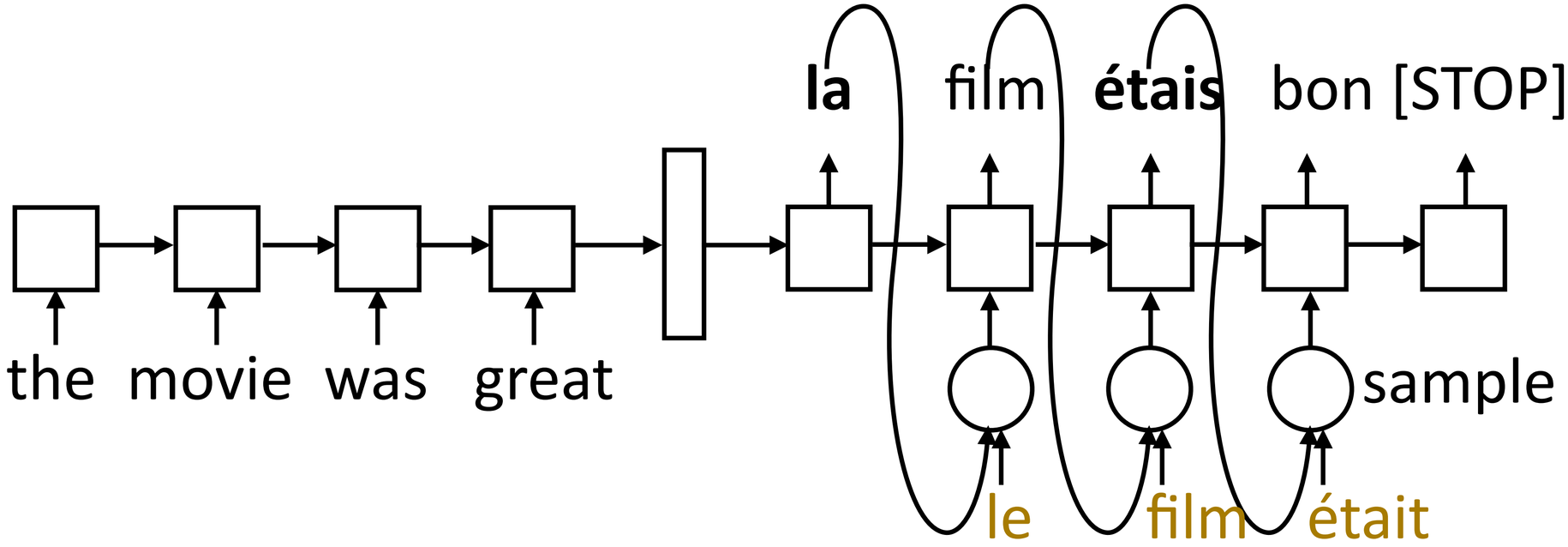
Training



- Objective: maximize $\sum_{(\mathbf{x},\mathbf{y})} \sum_{i=1}^{n} \log P(y_i^* | \mathbf{x}, y_1^*, \ldots, y_{i-1}^*)$
- One loss term for each target-sentence word, feed the correct word regardless of model's prediction (called "teacher forcing")

Training: Scheduled Sampling

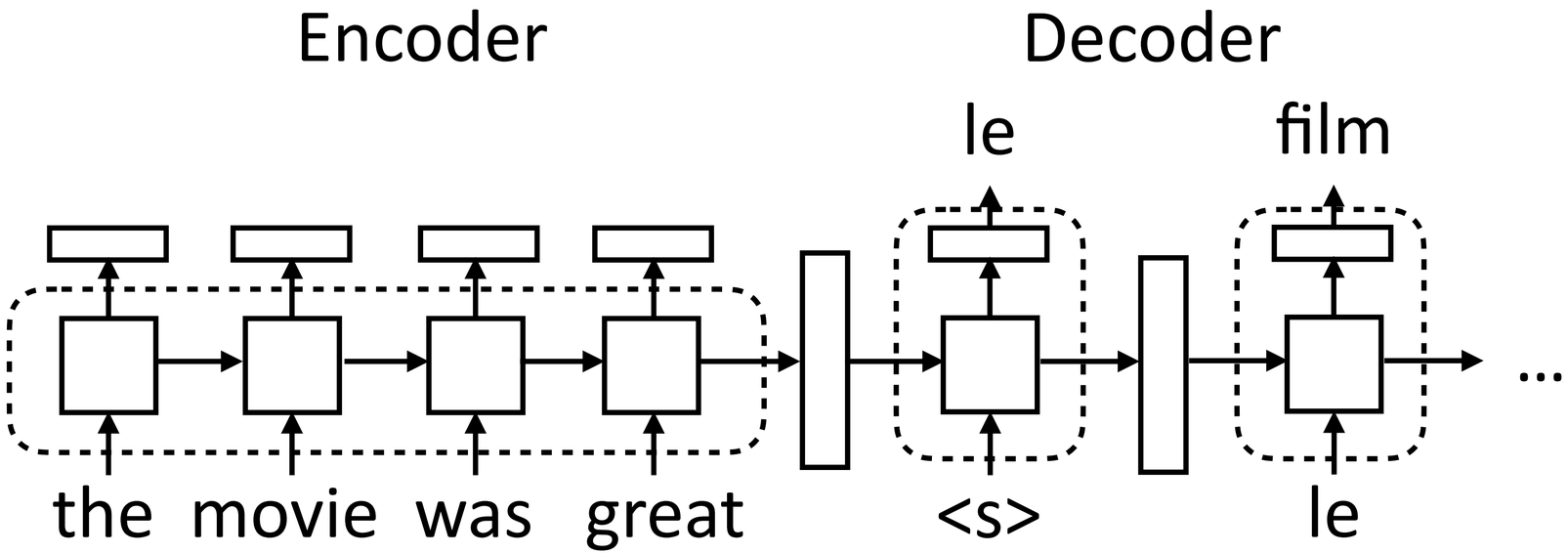
- Model needs to do the right thing even with its own predictions



- Scheduled sampling: with probability *p*, take the gold (human) translation as input, else take the model's prediction
- Starting with *p* = 1 and decaying it works best

Bengio et al. (2015)

Implementing seq2seq Models



- Encoder: consumes sequence of tokens, produces a vector. Analogous to encoders for classification/tagging tasks $P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) = \text{softmax}(W\bar{h}_i)$
- Decoder: separate module, single cell. Takes two inputs: hidden state (vector *h* or tuple (*h*, *c*)) and previous token. Outputs token + new state

Implementation Details

- Sentence lengths vary for both encoder and decoder:
  - Typically pad everything to the right length
- Encoder: Can be a LSTM/CNN/Transformer…
- Batching is a bit tricky:
  - encoder should use pack_padded_sequence to handle different lengths.
  - The decoder should pad everything to the same length and use a mask to only accumulate “valid” loss terms
  - Label vectors may look like [num timesteps x batch size x num labels]

Implementation Details (cont')

- Decoder: execute one step of computation at a time, so computation graph is formulated as taking one input + hidden state.
  - Test time: do this until you generate the [STOP] token
  - Training time: do this until you reach the gold stopping point

- Beam search (next class): can help with lookahead. Finds the (approximate) highest scoring sequence:

$$\text{argmax}_{\mathbf{y}} \prod_{i=1}^{n} P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1})$$

Attention

Problems with Seq2seq Models

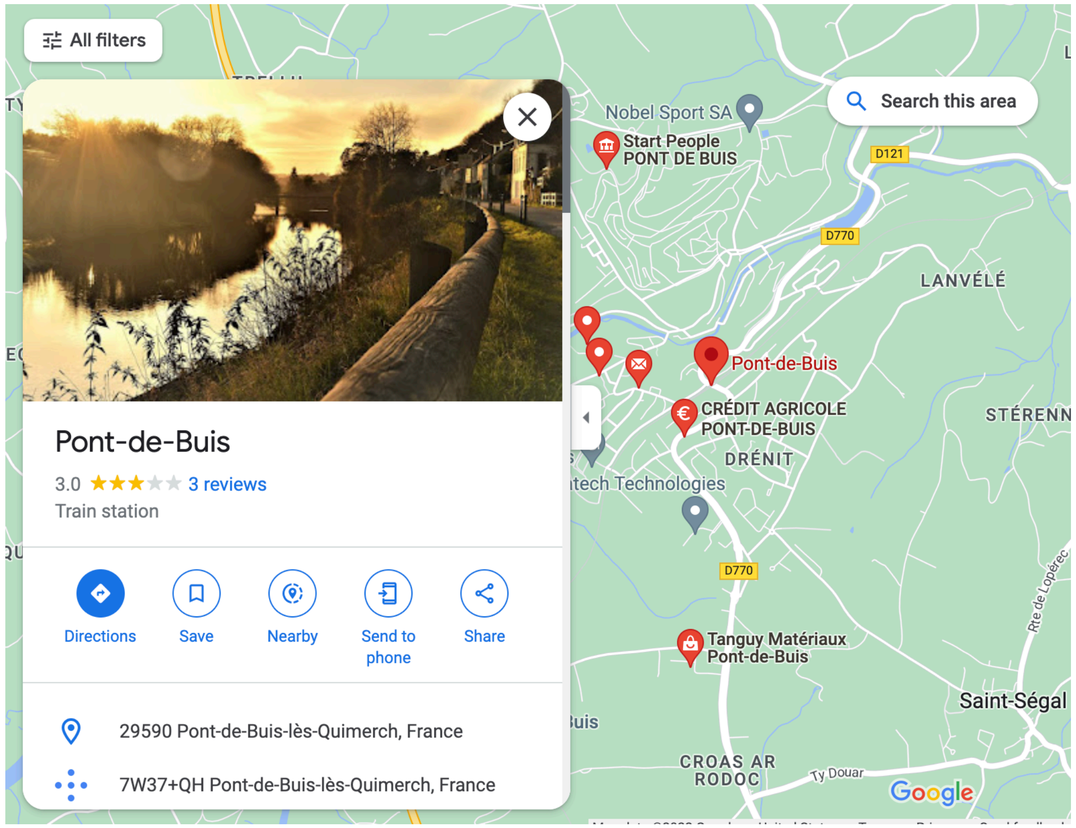
- Encoder-decoder models like to repeat themselves:

Un garçon joue dans la neige → *A boy plays in the snow* ***boy plays boy plays***

- Often a byproduct of training these models poorly. Input is forgotten by the LSTM so it gets stuck in a "loop" of generation the same output tokens again and again.
- Need some notion of input coverage or what input words we've translated

Rare/Unknown Words

The ecotax portico in Pont-de-Buis, around which a violent demonstration against the tax took place on Saturday, was taken down on Thursday morning.



Problems with Seq2seq Models

- Unknown words:

en: The *ecotax* portico in *Pont-de-Buis* , . . . [truncated] . . . , was taken down on Thursday morning

fr: Le *portique* *écotaxe* de *Pont-de-Buis* , . . . [truncated] . . . , a été *démonté* jeudi matin

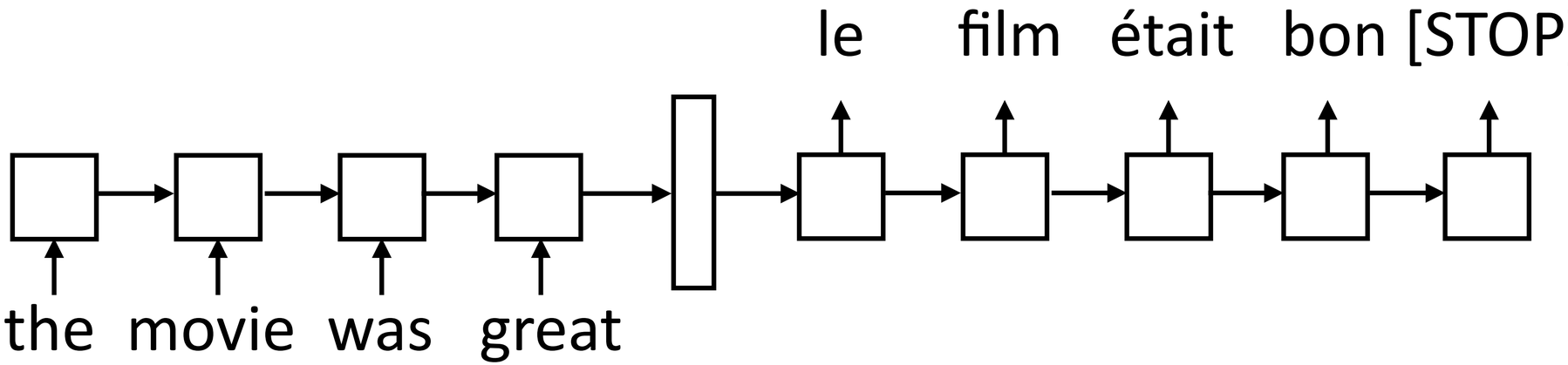
nn: Le `unk` de `unk` à `unk` , . . . [truncated] . . . , a été pris le jeudi matin

- Encoding these rare words into a vector space is really hard
- In fact, we don't want to encode them, we want a way of directly looking back at the input and copying them (Pont-de-Buis)

Jean et al. (2015), Luong et al. (2015)

Encoder-Decoder (Recap)

- Encode a sequence into a fixed-sized vector

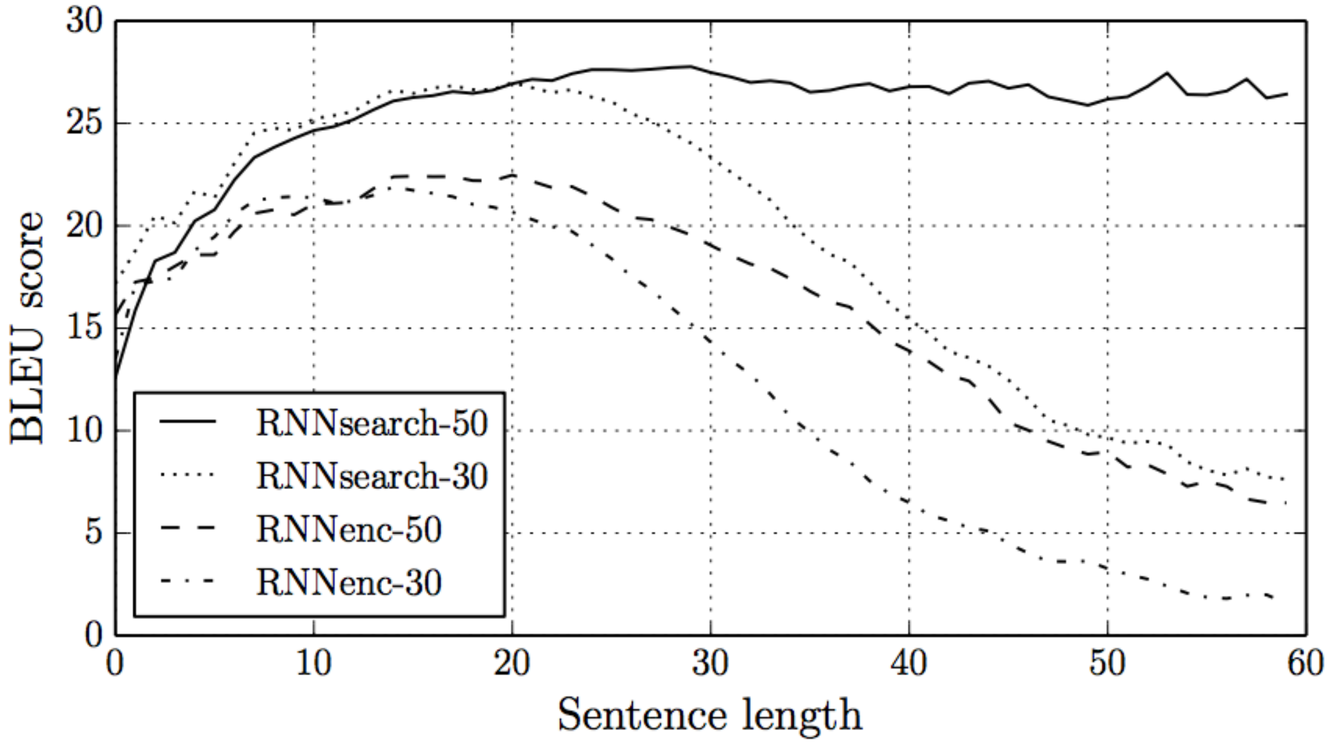


- Now use that vector to produce a series of tokens as output from a separate LSTM *decoder*
- Machine translation, NLG, summarization, dialog, and many other tasks (e.g., semantic parsing, syntactic parsing) can be done using this framework.

Sutskever et al. (2014)

Problems with Seq2seq Models

- Bad at long sentences: 1) a fixed-size hidden representation doesn't scale; 2) LSTMs still have a hard time remembering for really long sentences



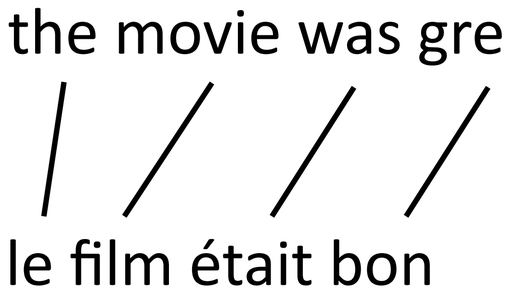
RNNenc: the model we've discussed so far

RNNsearch: uses attention

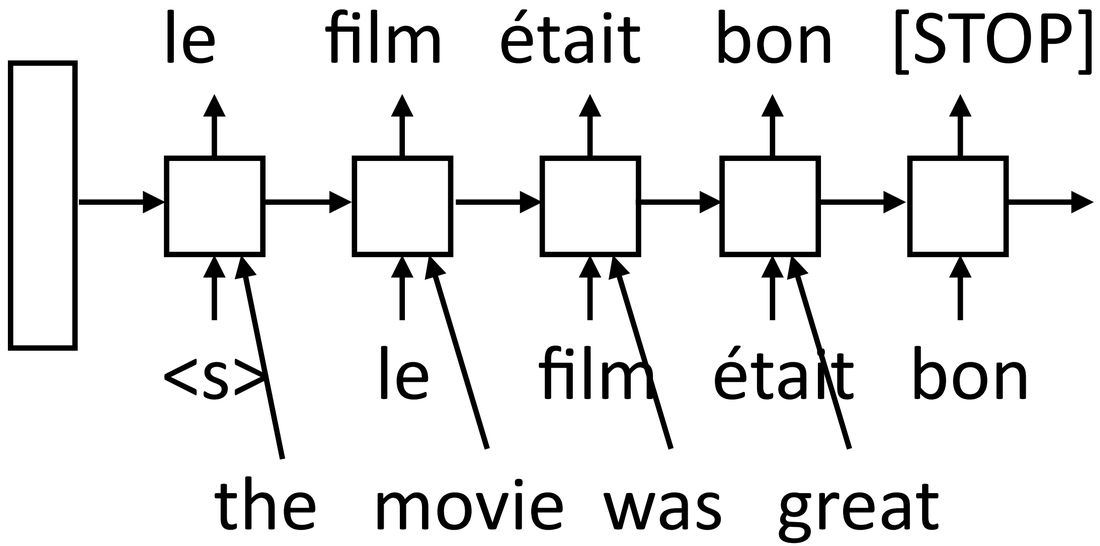
Bahdanau et al. (2014)

Aligned Inputs

- Suppose we knew the source and target would be word-by-word translated (recall the word alignment we talked about in phrase-based MT)

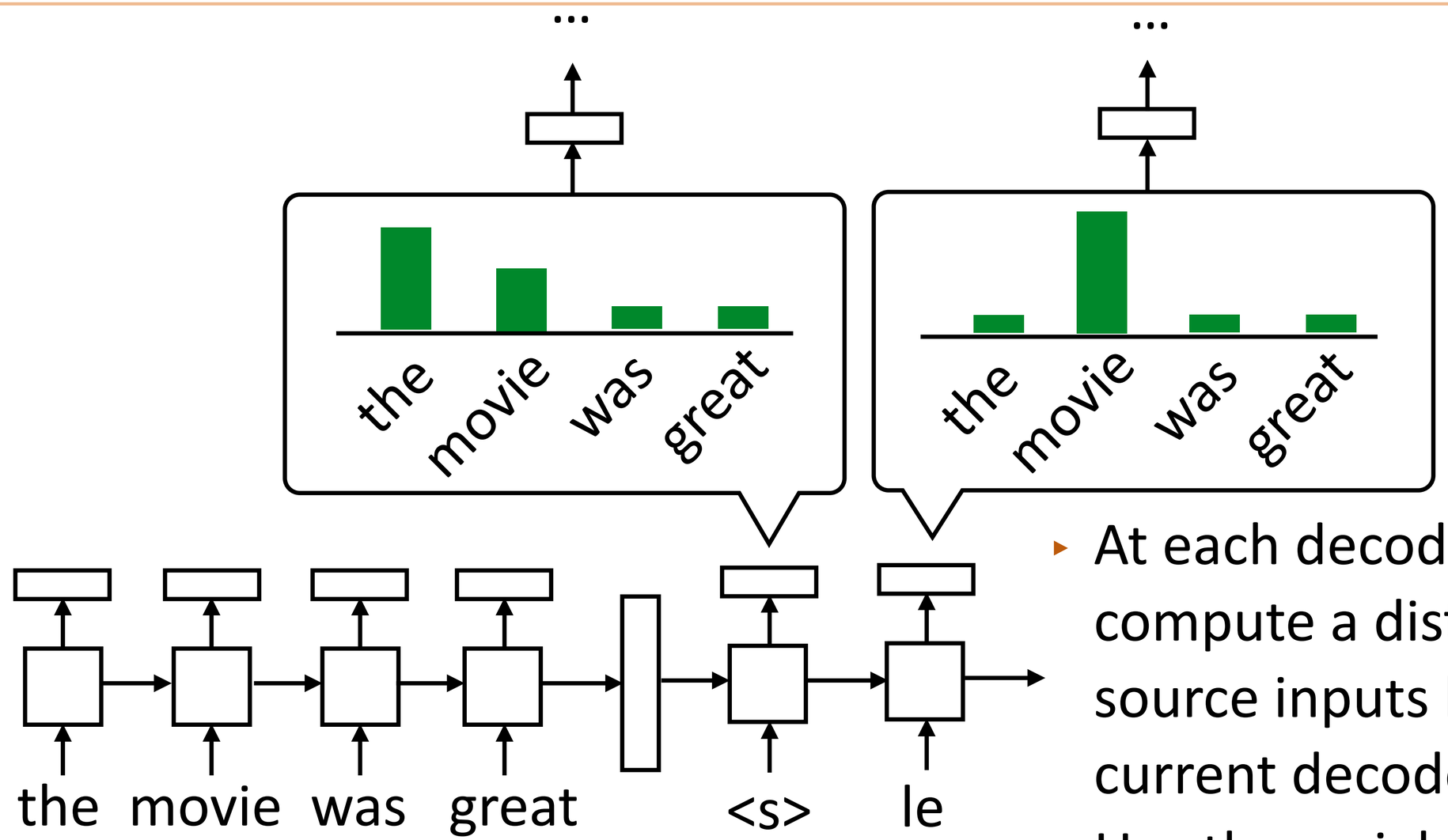


- Can look at the corresponding input word when translating — this could scale!
- Less burden on the hidden states



- How can we achieve this without hardcoding it?

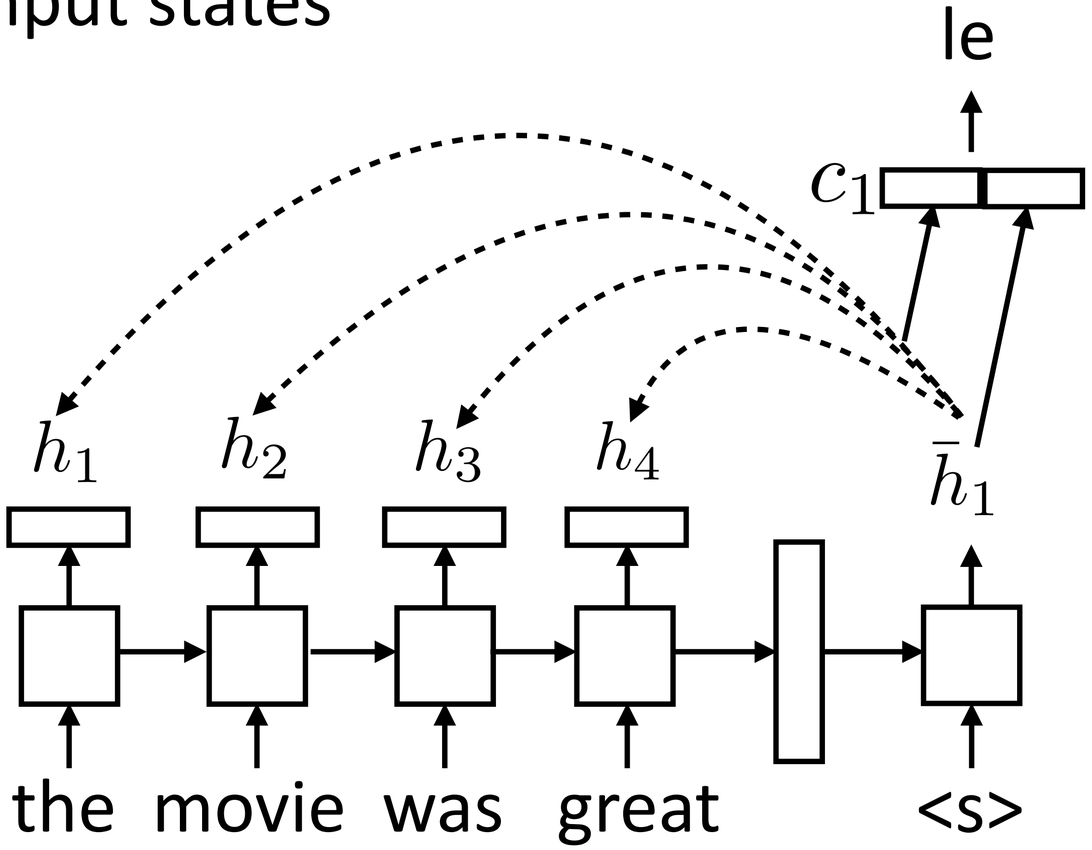
Attention



- At each decoder state, compute a distribution over source inputs based on current decoder state
- Use the weighted sum of input tokens to predict output

Attention

- For each decoder state, compute weighted sum of input states



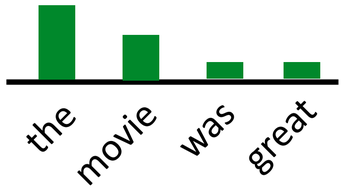
- No attn: $P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) = \text{softmax}(W\bar{h}_i)$

$$P(y_i|\mathbf{x}, y_1, \ldots, y_{i-1}) = \text{softmax}(W[c_i; \bar{h}_i])$$

$$c_i = \sum_j \alpha_{ij} h_j$$

- Weighted sum of input hidden states (vector)

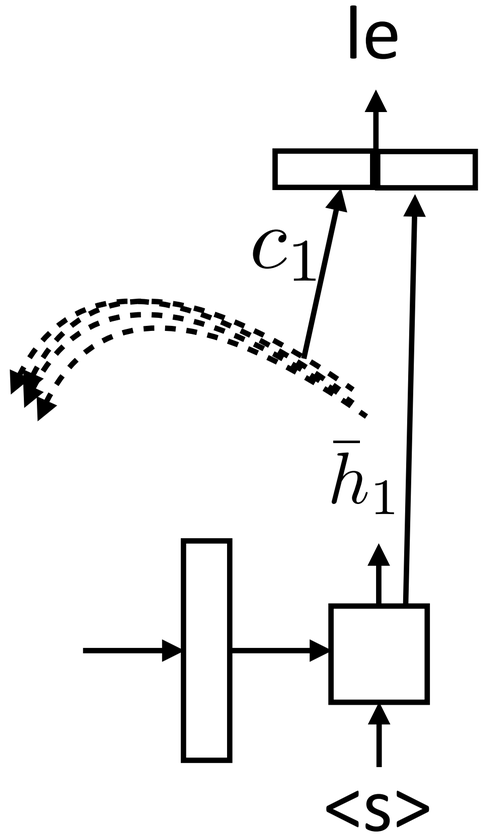
$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})}$$



$$e_{ij} = f(\bar{h}_i, h_j)$$

- Some function f (e.g., dot product)

Attention



$$c_i = \sum_j \alpha_{ij} h_j$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})}$$

$$e_{ij} = f(\bar{h}_i, h_j)$$

$$f(\bar{h}_i, h_j) = \tanh(W[\bar{h}_i; h_j])$$

- Bahdanau+ (2014): additive

$$f(\bar{h}_i, h_j) = \bar{h}_i \cdot h_j$$

- Luong+ (2015): dot product

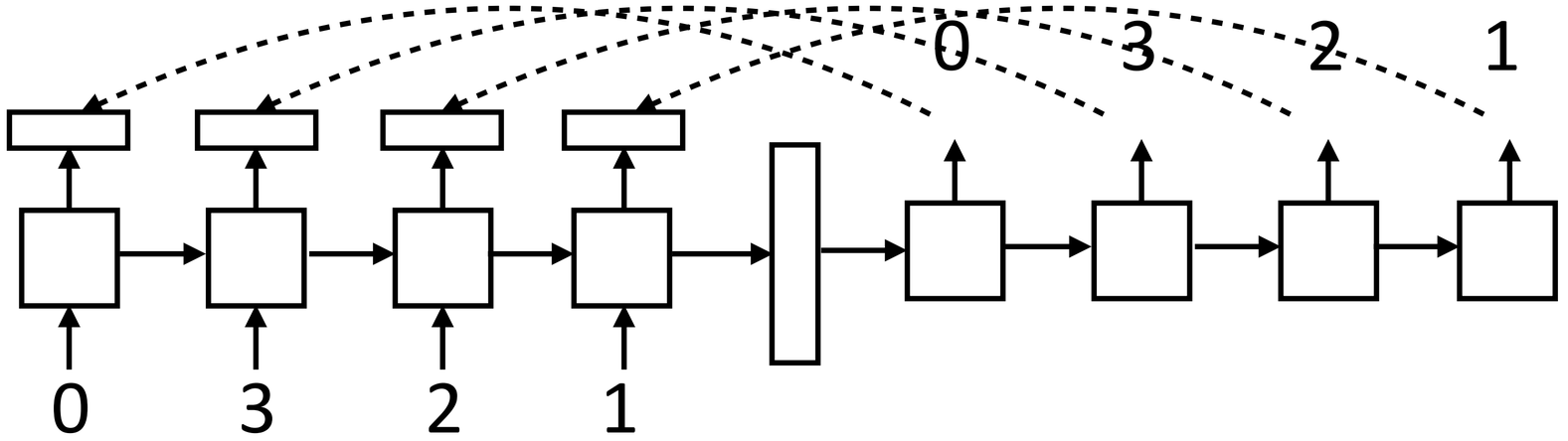
$$f(\bar{h}_i, h_j) = \bar{h}_i^\top W h_j$$

- Luong+ (2015): bilinear

- Note that this all uses outputs of hidden layers

What can attention do?

- Learning to copy — how might this work?



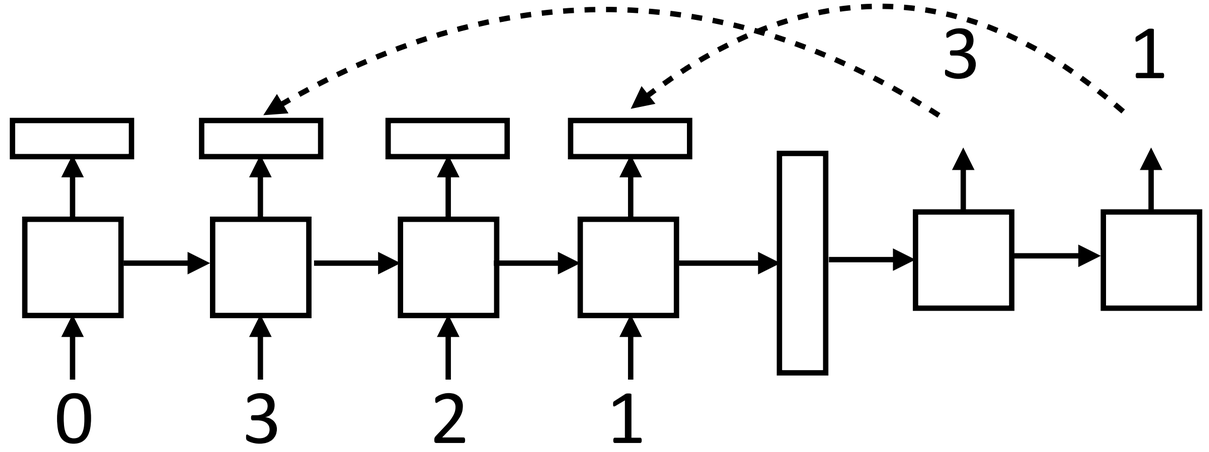
- LSTM can learn to count with the right weight matrix

- This is a kind of position-based addressing

Luong et al. (2015)

What can attention do?

- Learning to subsample tokens



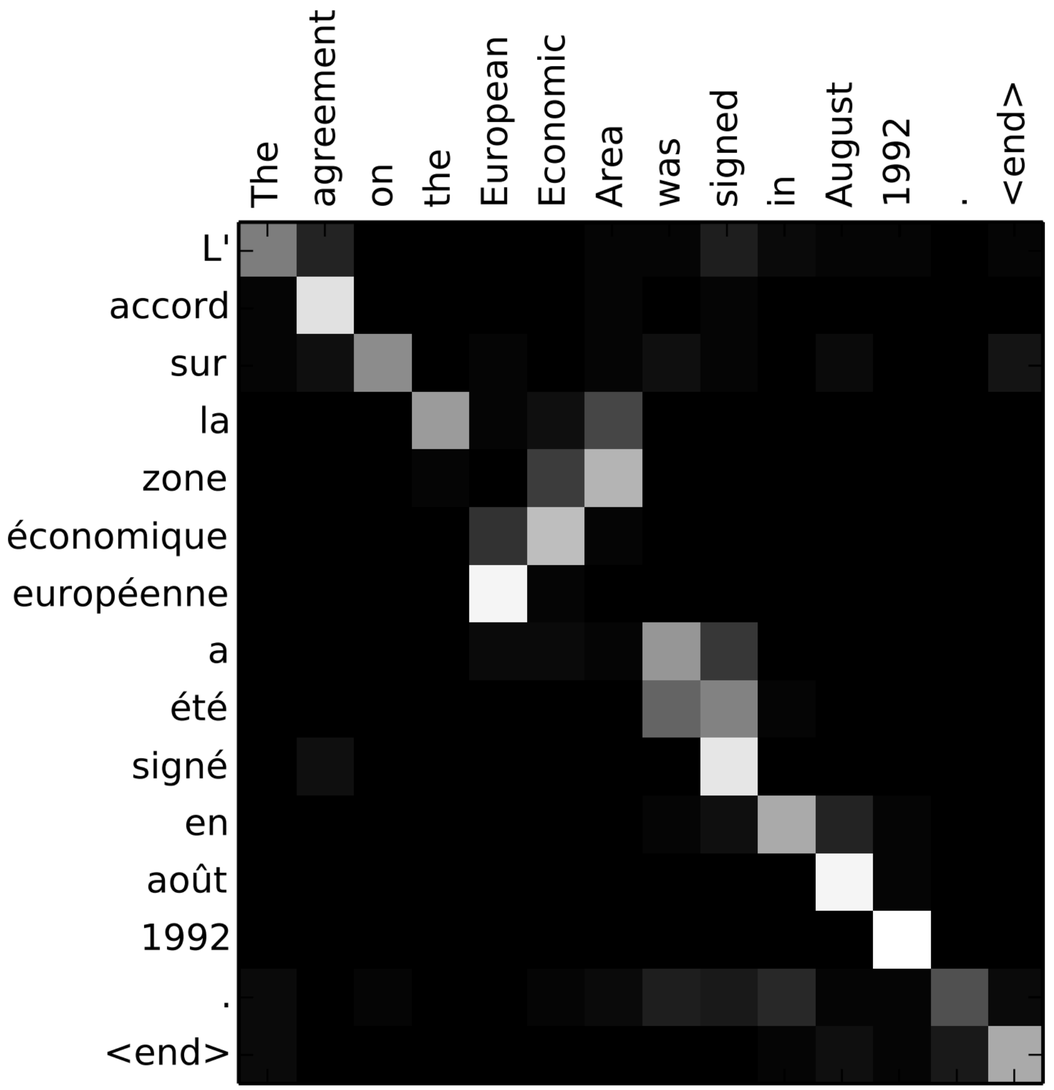
- Need to count (for ordering) and also determine which tokens are in/out

- Content-based addressing

Luong et al. (2015)

Attention

- Encoder hidden states capture contextual source word identity (“soft” word alignment)
- Decoder hidden states are now mostly responsible for selecting what to attend to
- Doesn’t take a complex hidden state to walk monotonically through a sentence and spit out word-by-word translations

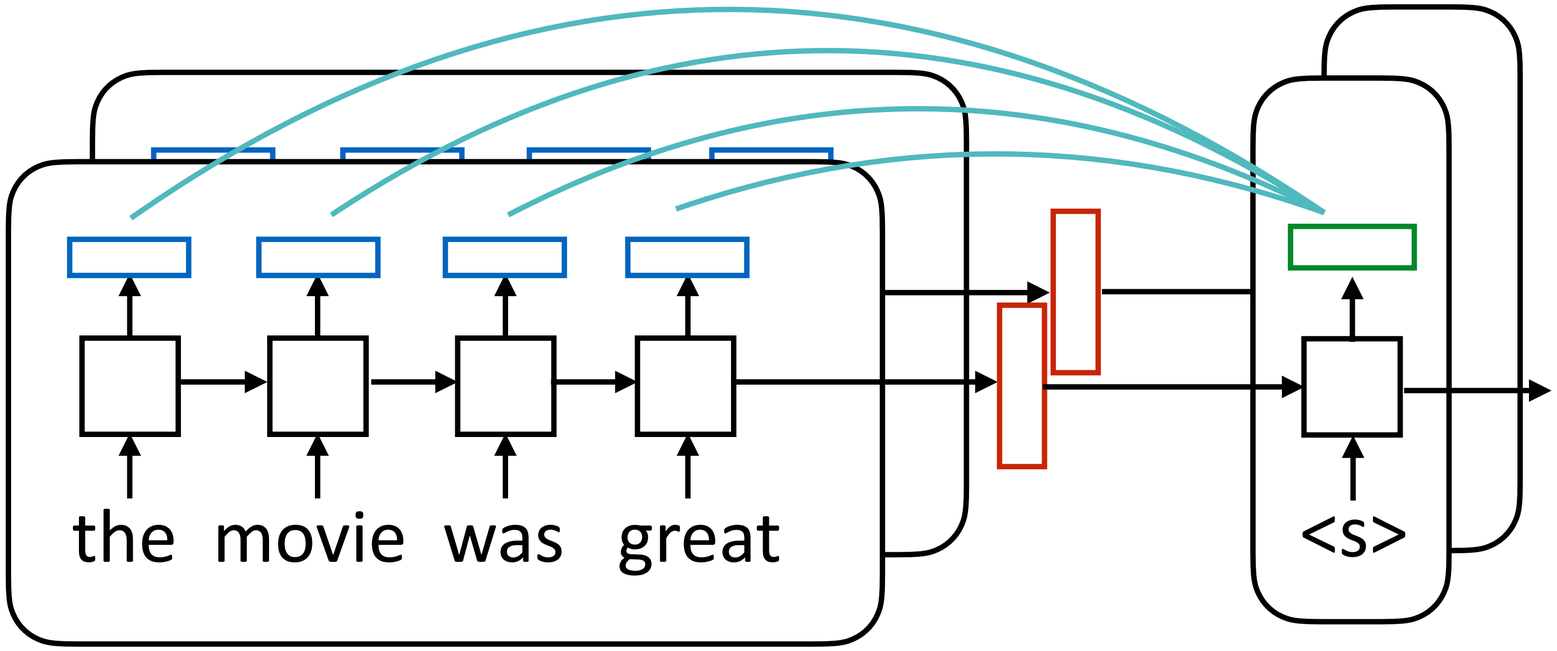


α_{ij}

Bahdanau et al. (2014)

Batching Attention

token outputs: batch size x sentence length x hidden size



hidden state: batch size x hidden size

$$e_{ij} = f(\bar{h}_i, h_j)$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})}$$

sentence outputs: batch size x hidden size

attention scores = batch size x sentence length

c = batch size x hidden size

$$c_i = \sum_j \alpha_{ij} h_j$$

- Make sure tensors are the right size!

Luong et al. (2015)

Some MT Results

“Early” Neural MT

Effective Approaches to Attention-based Neural Machine Translation

Minh-Thang Luong **Hieu Pham** **Christopher D. Manning**
Computer Science Department, Stanford University, Stanford, CA 94305
{lmthang,hyhieu,manning}@stanford.edu

Abstract

An attentional mechanism has lately been used to improve neural machine translation (NMT) by selectively focusing on parts of the source sentence during translation. However, there has been little work exploring useful architectures for attention-based NMT. This paper examines two simple and effective classes of attentional mechanism: a *global* approach which always attends to all source words and a *local* one that only looks at a subset of source words at a time. We demonstrate the effectiveness of both approaches on the

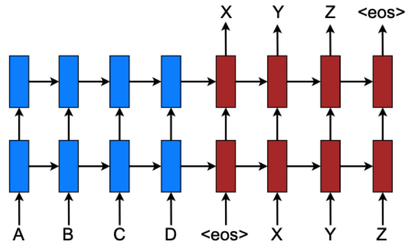


Figure 1: **Neural machine translation** – a stacking recurrent architecture for translating a source sequence A B C D into a target sequence X Y Z. Here, <eos> marks the end of a sentence.

ing plain SGD, (c) a simple learning rate schedule is employed – we start with a learning rate of 1; after 5 epochs, we begin to halve the learning rate every epoch, (d) our mini-batch size is 128, and (e) the normalized gradient is rescaled whenever its norm exceeds 5. Additionally, we also use dropout with probability 0.2 for our LSTMs as suggested by (Zaremba et al., 2015). For dropout models, we train for 12 epochs and start halving the learning rate after 8 epochs. For local attention models, we empirically set the window size $D = 10$.

Our code is implemented in MATLAB. When running on a single GPU device Tesla K40, we achieve a speed of 1K *target* words per second. It takes 7–10 days to completely train a model.

- TensorFlow first released in Nov 2015.
- PyTorch first released in 2016.

Luong et al. (2015)

MT Examples

src	In einem Interview sagte Bloom jedoch , dass er und Kerr sich noch immer lieben .
ref	However , in an interview , Bloom has said that he and *Kerr* still love each other .
best	In an interview , however , Bloom said that he and *Kerr* still love .
base	However , in an interview , Bloom said that he and **Tina** were still <unk> .

- best = with attention, base = no attention
- NMT systems can hallucinate words, especially when not using attention — phrase-based doesn't do this

Luong et al. (2015)

MT Examples

src	Wegen der von Berlin und der Europäischen Zentralbank verhängten strengen Sparpolitik in Verbindung mit der Zwangsjacke , in die die jeweilige nationale Wirtschaft durch das Festhalten an der gemeinsamen Währung genötigt wird , sind viele Menschen der Ansicht , das Projekt Europa sei zu weit gegangen
ref	The *austerity imposed by Berlin and the European Central Bank , coupled with the straitjacket* imposed on national economies through adherence to the common currency , has led many people to think Project Europe has gone too far .
best	Because of the strict *austerity measures imposed by Berlin and the European Central Bank in connection with the straitjacket* in which the respective national economy is forced to adhere to the common currency , many people believe that the European project has gone too far .
base	Because of the pressure **imposed by the European Central Bank and the Federal Central Bank with the strict austerity** imposed on the national economy in the face of the single currency , many people believe that the European project has gone too far .

- best = with attention, base = no attention

Luong et al. (2015)

Results: WMT English-French

- 12M sentence pairs

Classic phrase-based system: ~**33** BLEU, uses additional target-language data

Rerank with LSTMs: **36.5** BLEU (long line of work here; Devlin+ 2014)

Sutskever+ (2014) seq2seq single: **30.6** BLEU (input reversed)

Sutskever+ (2014) seq2seq ensemble: **34.8** BLEU

Luong+ (2015) seq2seq ensemble with attention and rare word handling: **37.5** BLEU

- But English-French is a really easy language pair and there's *tons* of data for it! Does this approach work for anything harder?

Results: WMT English-German

- 4.5M sentence pairs

Classic phrase-based system: **20.7** BLEU

Luong+ (2014) seq2seq: **14** BLEU

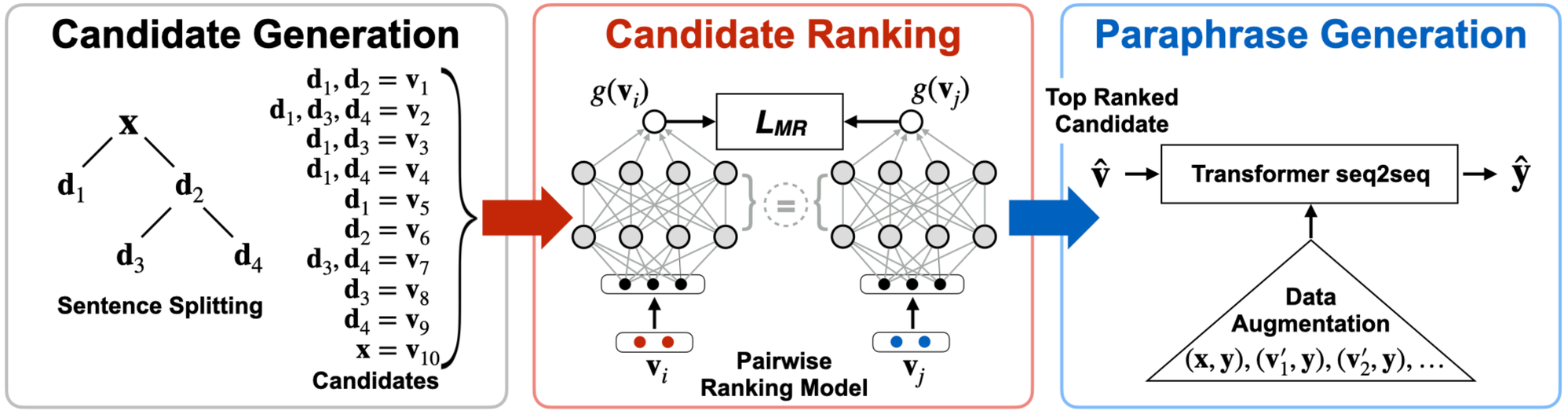
Luong+ (2015) seq2seq ensemble with rare word handling: **23.0** BLEU

- Not nearly as good in absolute BLEU, but not really comparable across languages
- French, Spanish = easiest
  German, Czech = harder
  Japanese, Russian = hard (grammatically different, lots of morphology…)

Other Applications of Seq2Seq

Text-to-Text Generation

- Text Simplification (with readability constraints)



Input sentece:

Since 2010, project researchers have uncovered documents in Portugal that have revealed who owned the ship

→ **seq2seq models** (RNN, Transformer) →

Generated Output:

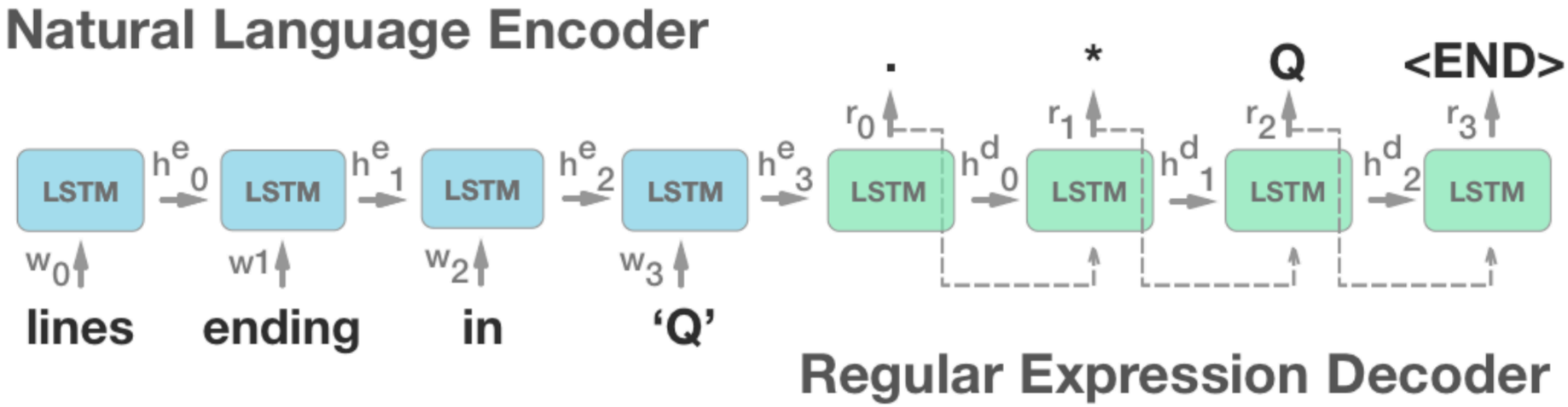
Scientists have found documents in Portugal.

They have also found out who owned the ship.

Mounica Maddela, Fernando Alva-Manchego, Wei Xu. “Controllable Text Simplification with Explicit Paraphrasing” (NAACL 2021)

Regex Prediction

- Seq2seq models can be used for many other tasks!
- Predict regex from text



- Problem: requires a lot of data: 10,000 examples needed to get ~60% accuracy on pretty simple regexes

Locascio et al. (2016)

Semantic Parsing as Translation

"what states border Texas"

↓

$$\lambda\ x\ \text{state}(\ x\ ) \wedge \text{borders}(\ x\ ,\ e89\ )$$

- Write down a linearized form of the semantic parse, train seq2seq models to directly translate into this representation
- No need to have an explicit grammar, simplifies algorithms
- Might not produce well-formed logical forms, might require lots of data

Jia and Liang (2015)

Semantic Parsing/Lambda Calculus: https://www.youtube.com/watch?v=OocGXG-BY6k&t=200s

SQL Generation

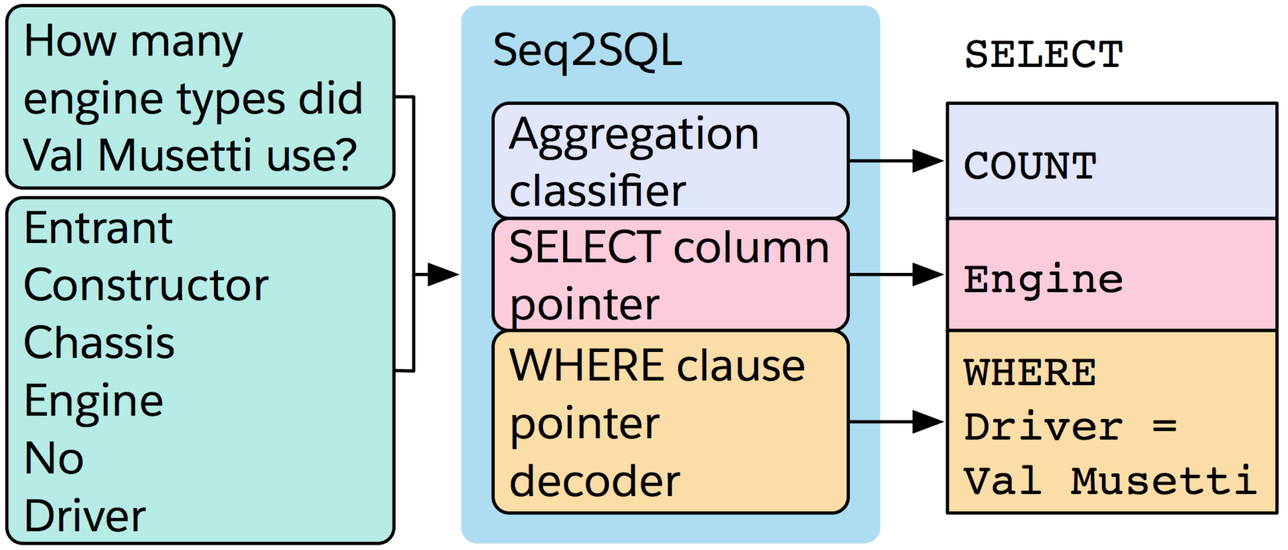
- Convert natural language description into a SQL query against some DB
- How to ensure that well-formed SQL is generated?
  - Three components
- How to capture column names + constants?
  - Pointer mechanisms

Question:

How many CFL teams are from York College?

SQL:

```
SELECT COUNT CFL Team FROM
CFLDraft WHERE College = “York”
```



Zhong et al. (2017)