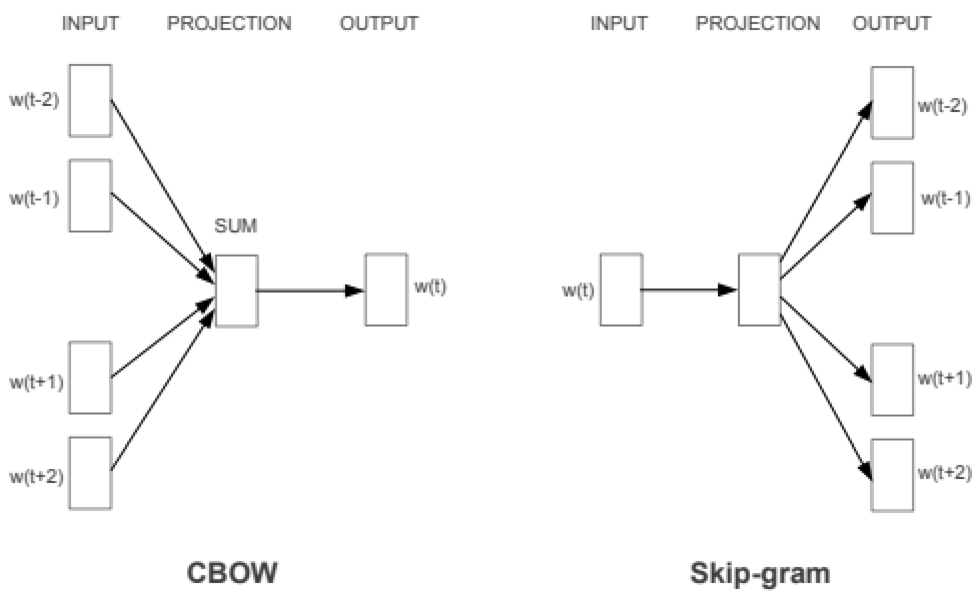
Neural nets for NLP

Neural Networks

WORD VECTORS

How to represent words?

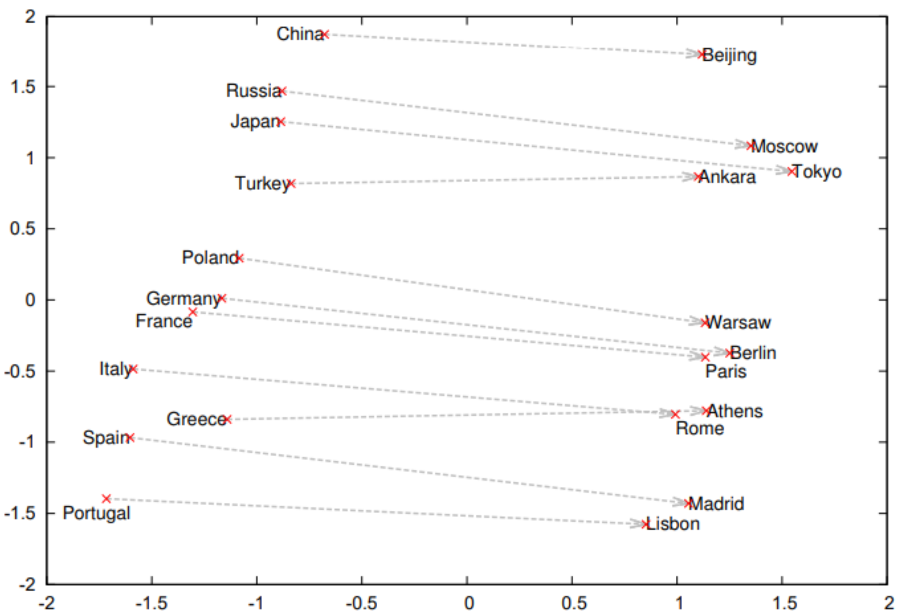
- We understand the meaning from the context
- Train a model to predict words based on context (or vice-versa)



- Words are discrete, but
  input layer assings a vector to each word.
  So does the output layer
- The words end up in *meaningful* positions

Word vector arithmetic

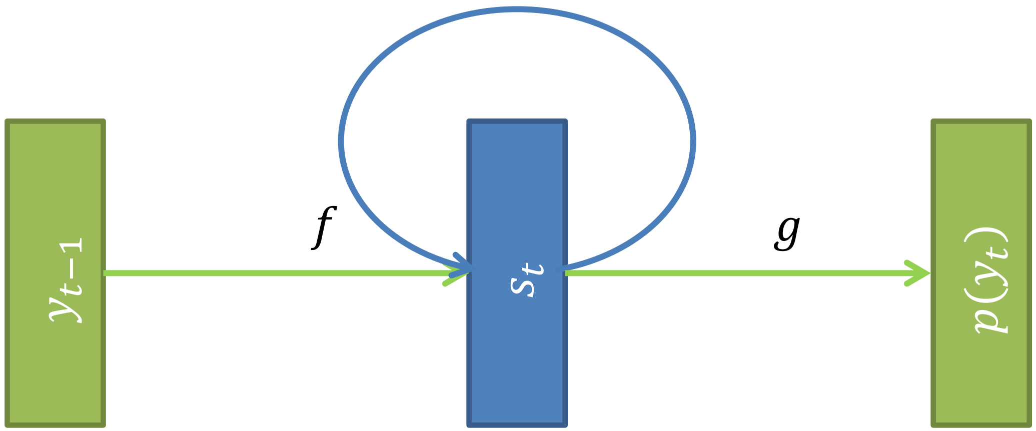
- King – queen ≈ man – woman
- Italy – Rome ≈ Poland - Warsaw



Type of relationship	Word Pair 1		Word Pair 2	
Common capital city	Athens	Greece	Oslo	Norway
All capital cities	Astana	Kazakhstan	Harare	Zimbabwe
Currency	Angola	kwanza	Iran	rial
City-in-state	Chicago	Illinois	Stockton	California
Man-Woman	brother	sister	grandson	granddaughter
Adjective to adverb	apparent	apparently	rapid	rapidly
Opposite	possibly	impossibly	ethical	unethical
Comparative	great	greater	tough	tougher
Superlative	easy	easiest	lucky	luckiest
Present Participle	think	thinking	read	reading
Nationality adjective	Switzerland	Swiss	Cambodia	Cambodian
Past tense	walking	walked	swimming	swam
Plural nouns	mouse	mice	dollar	dollars
Plural verbs	work	works	speak	speaks

ATTENTION MECHANISM

RNNs Learn $p(Y)$



Decompose

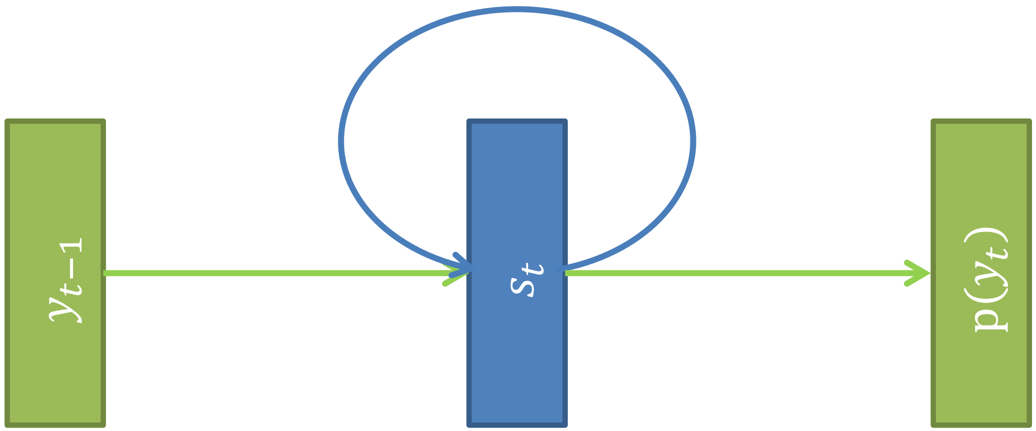
$$p(Y) = \prod p(y_t | y_{t-1}, y_{t-2}, \dots, y_1)$$

Model the probabilities using a recurrent relation

$$p(y_t | y_{t-1}, y_{t-2}, \dots, y_1) = g(s_t)$$
$$s_t = f(s_{t-1}, y_{t-1})$$

$g()$, $f()$ are implemented using neural networks, i.e. they are flexibly parameterized, smooth functions.

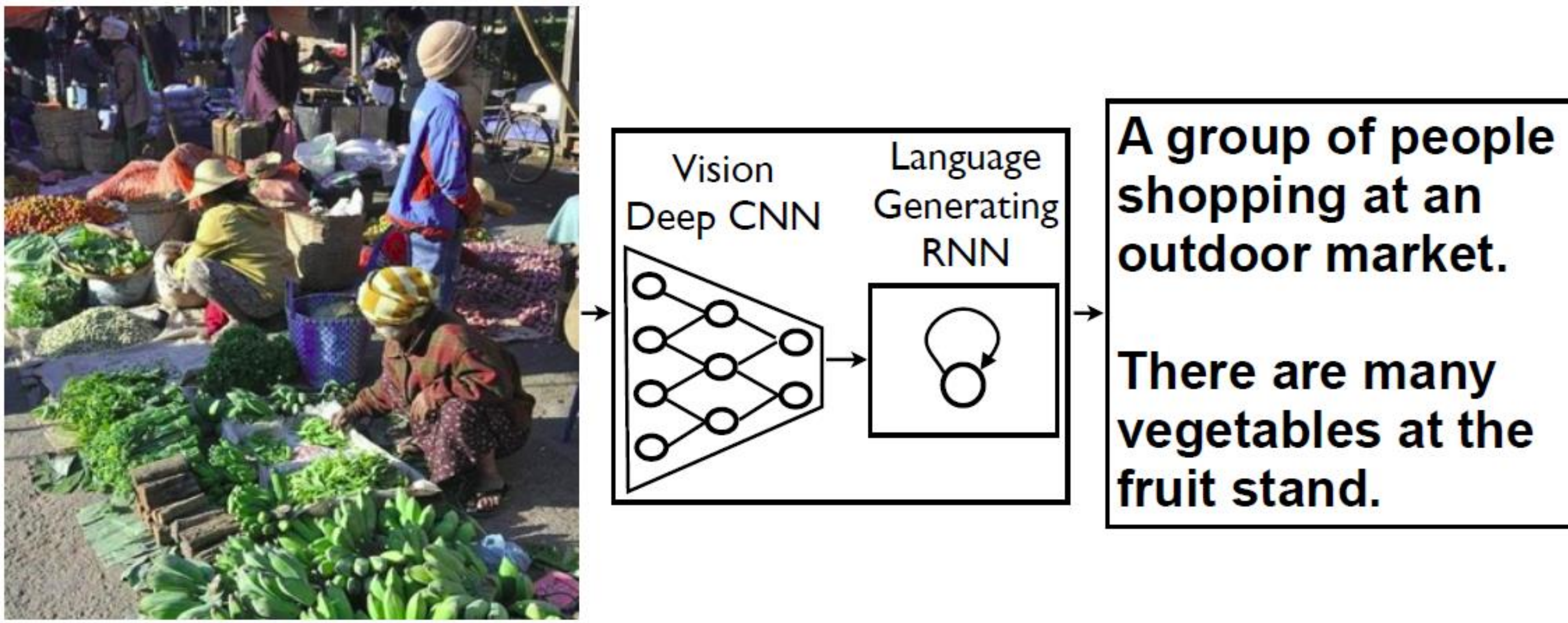
How to condition an RNN?



RNN gives us $p(Y)$ but we want $p(Y|X)$

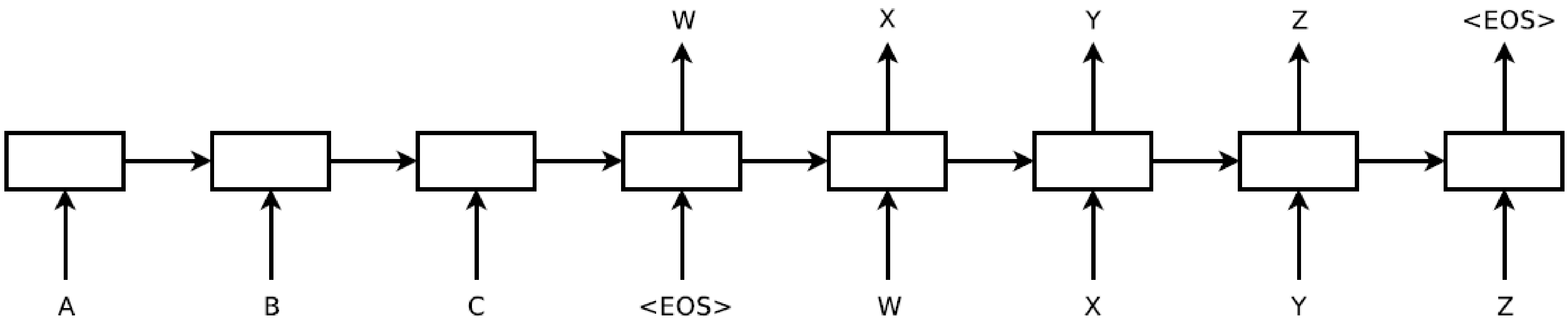
- Idea #1: conditioned through the first hidden state
- Idea #2: condition separately on every step

Idea #1
condition through the 1st hidden state



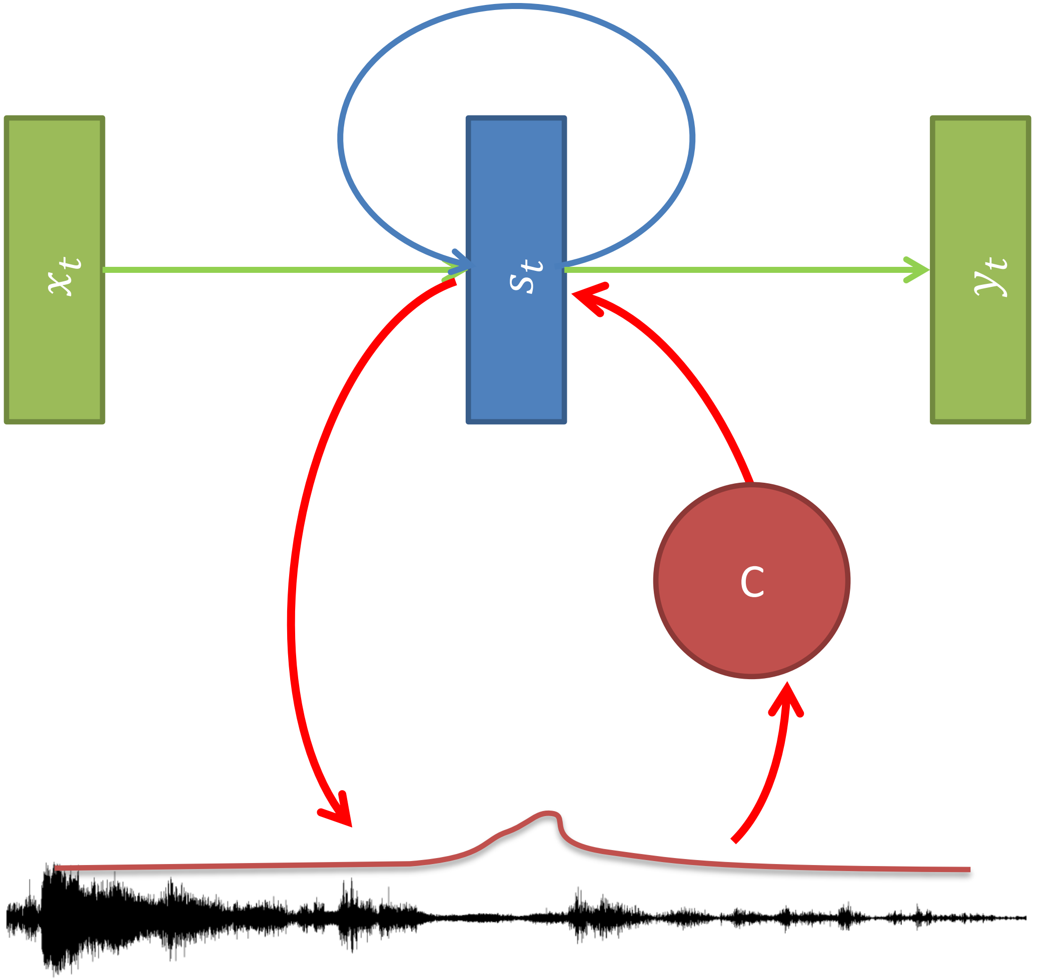
O. Vinyals et al, “Show and Tell: a Neural Image Caption Generator https://arxiv.org/abs/1411.4555

Idea #1
condition through the 1st hidden state



http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

Idea #2: Attention



1. Choose relevant frames

$$e_f = \text{score}\left(x_f, s_{t-1}\right)$$
$$\alpha_f = \text{SoftMax}(e)_f$$

2. Summarize into context

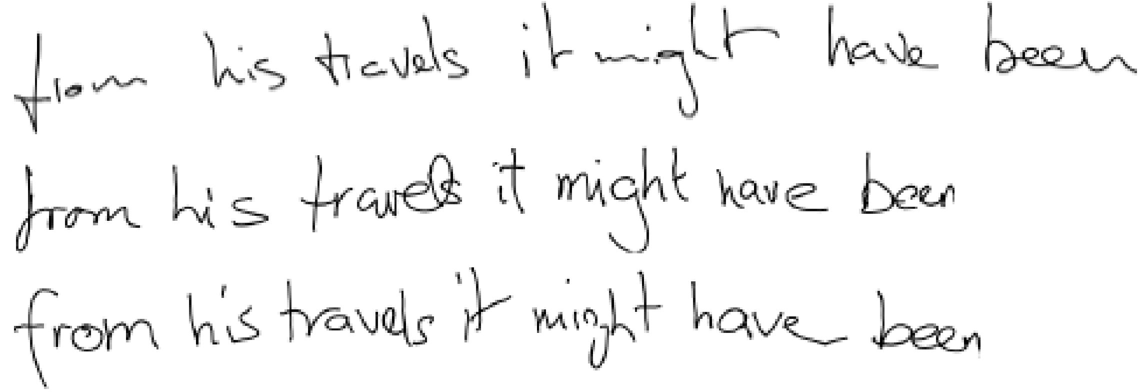
$$c = \sum_f \alpha_f x_f$$

3. Compute next state

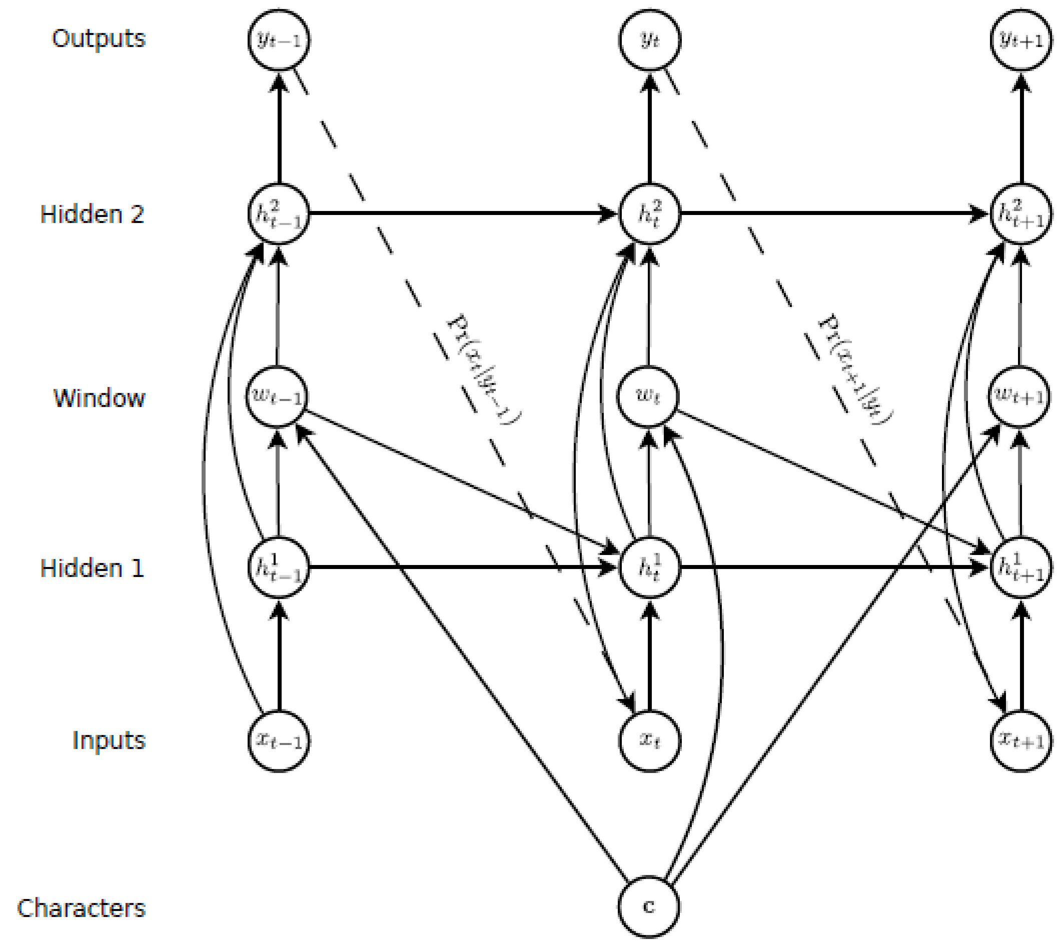
$$s_t = f\left(s_{t-1}, y_{t-1}, c\right)$$

Soft attention originally proposed for the Montreal neural translation project [Bahdanau2014]

Attention mechanism in RNNs

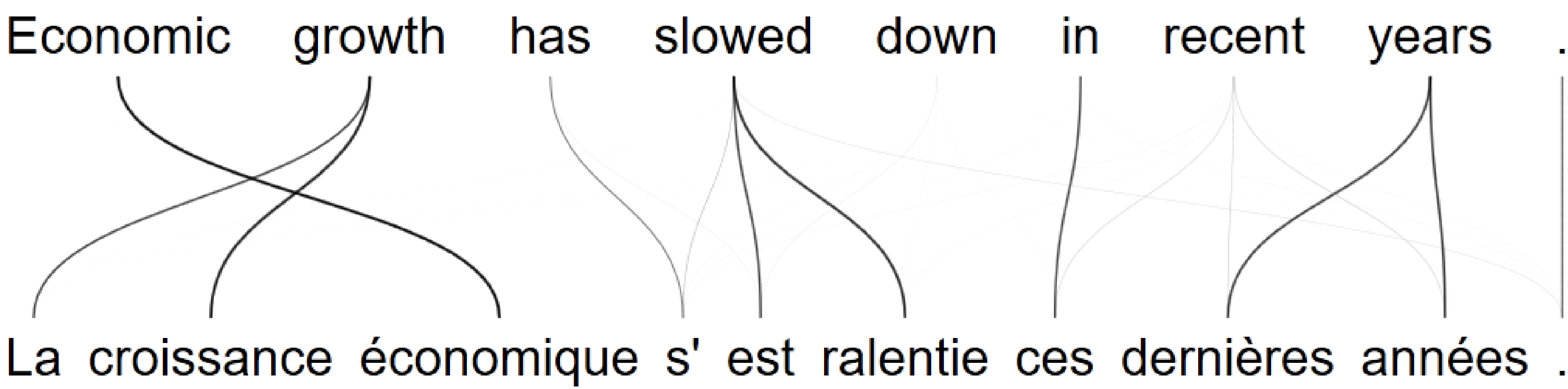


- This is a network to generate handwriting
- At each step the network looks at a *context* $\boldsymbol{c}$
- $\boldsymbol{c}$ is a summarization of a small fragment of the input sequence

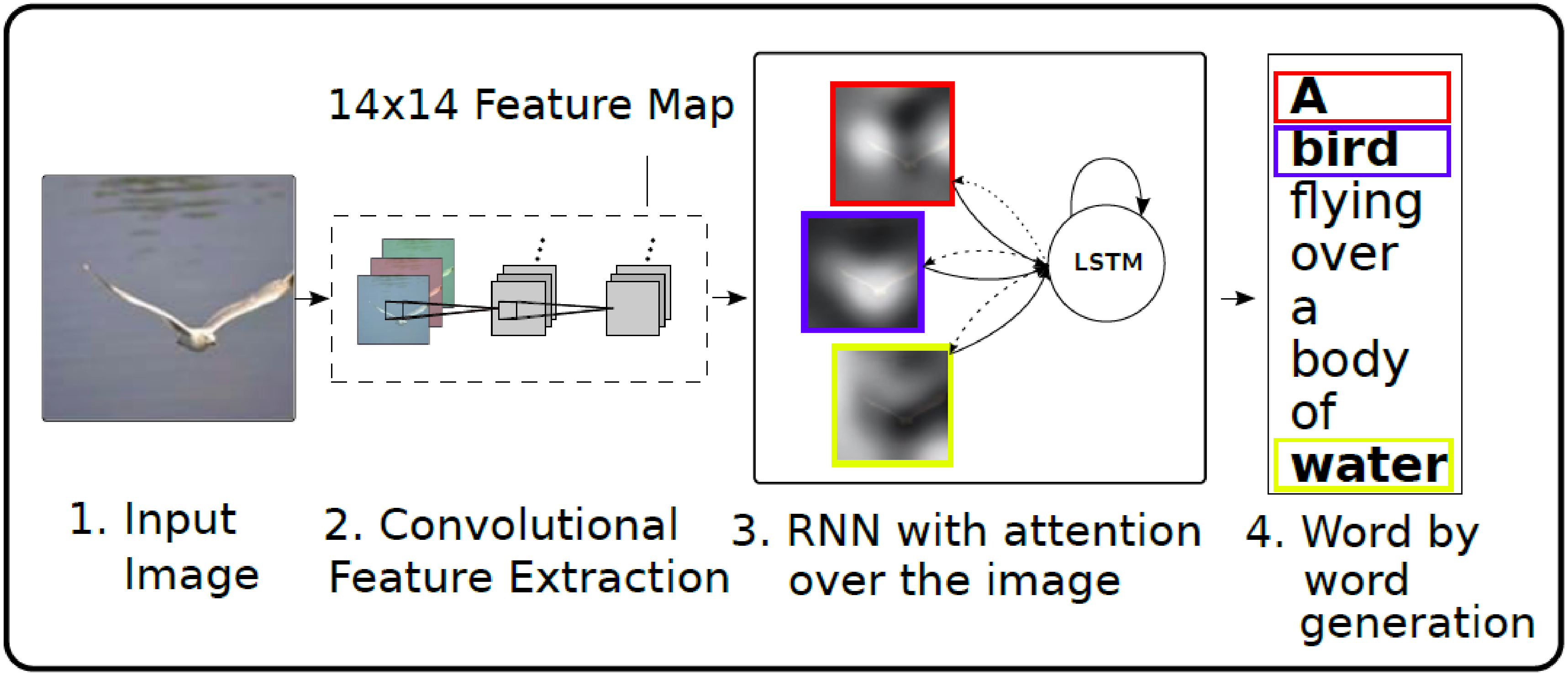


Graves, A., 2013. Generating Sequences With Recurrent Neural Networks. arXiv:1308.0850 [cs]

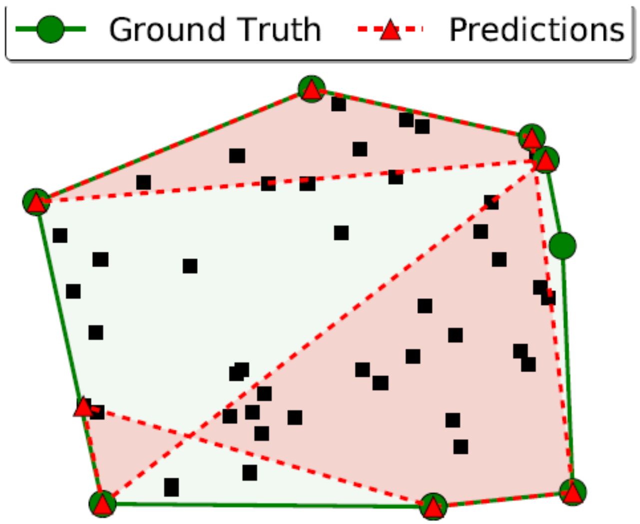
Attention mechanism in translation



Attention mechanism for captioning

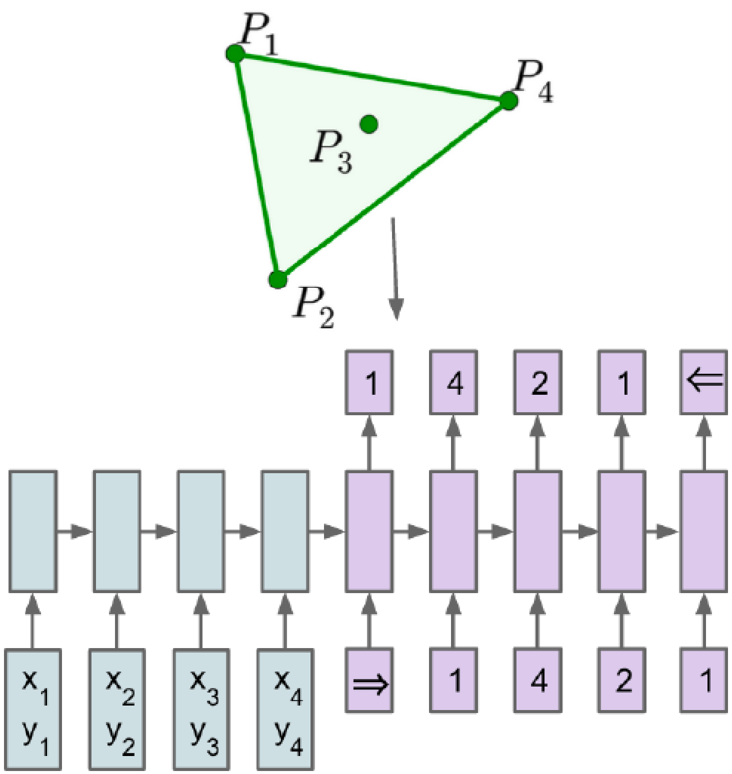


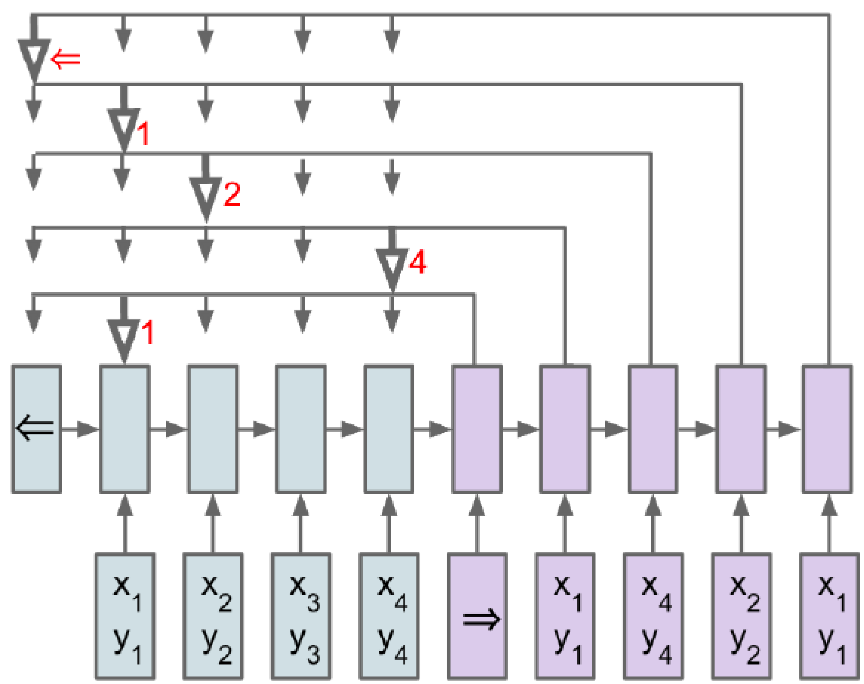
http://arxiv.org/pdf/1502.03044.pdf



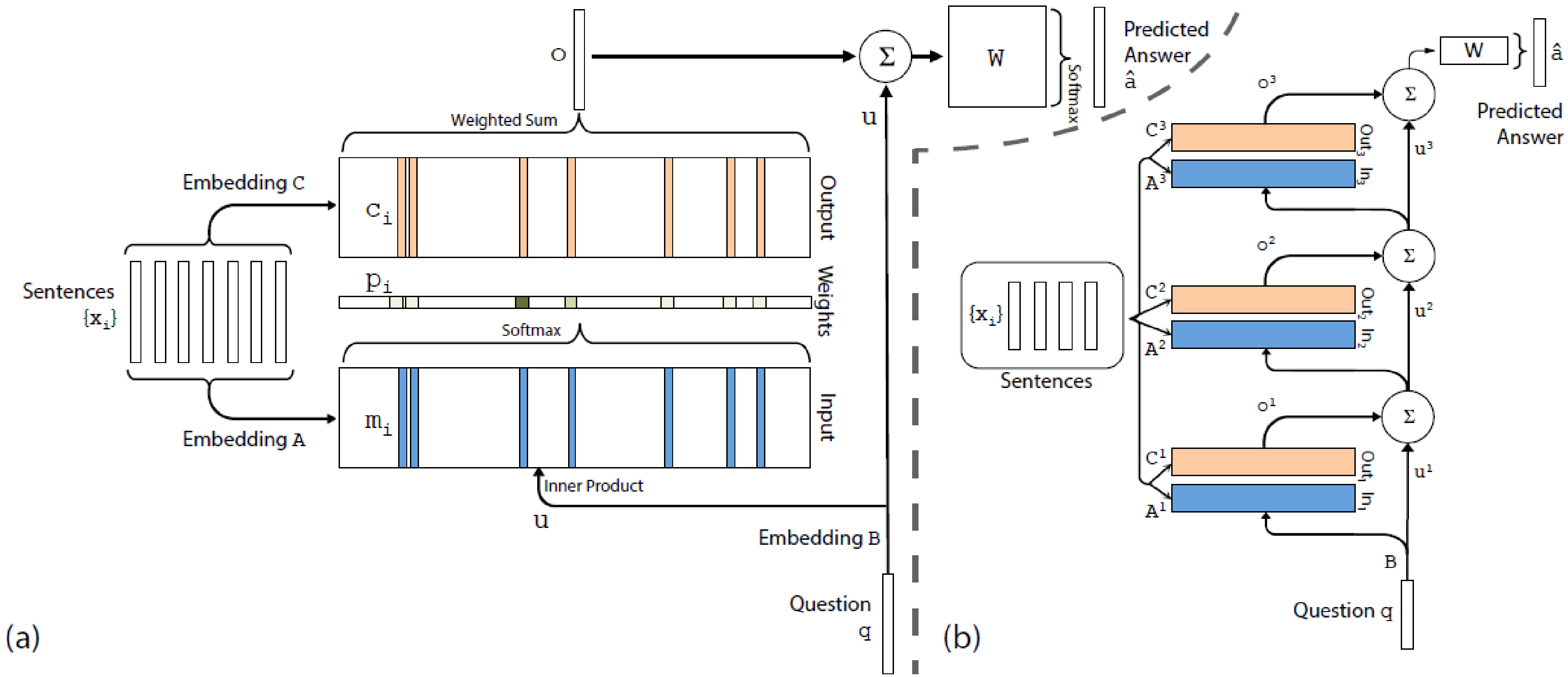
Convex Hulls & TSP

http://papers.nips.cc/paper/5866-pointer-networks.pdf





Reasoning – facts in memory



Story (16: basic induction)	Support	Hop 1	Hop 2	Hop 3
Brian is a frog.	yes	0.00	0.98	0.00
Lily is gray.		0.07	0.00	0.00
Brian is yellow.	yes	0.07	0.00	1.00
Julius is green.		0.06	0.00	0.00
Greg is a frog.	yes	0.76	0.02	0.00
What color is Greg? Answer: yellow		**Prediction: yellow**		

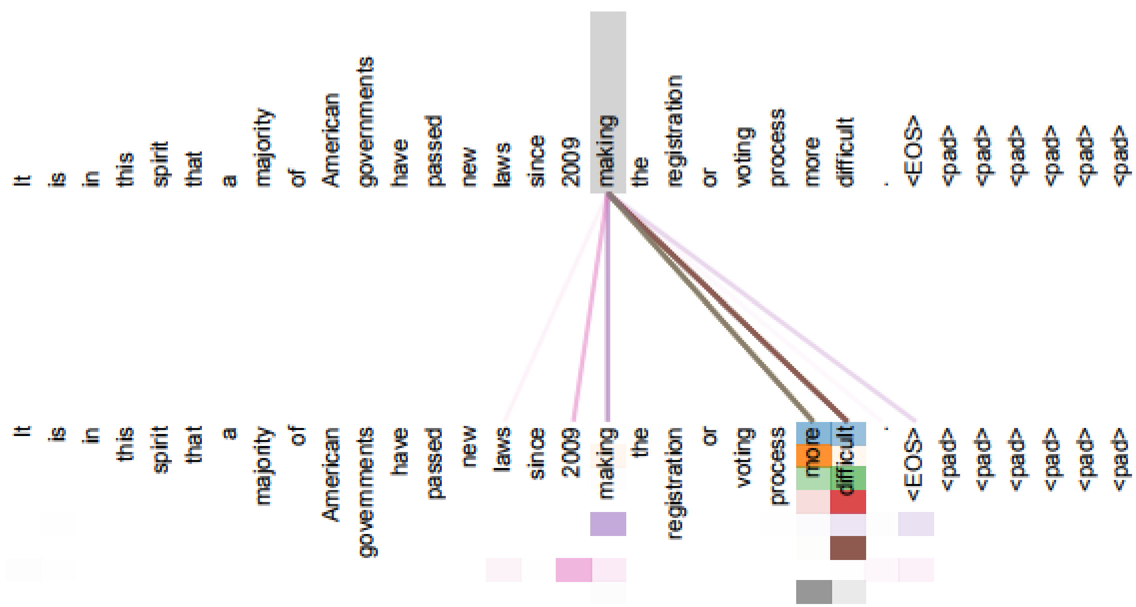
http://papers.nips.cc/paper/5846-end-to-end-memory-networks.pdf

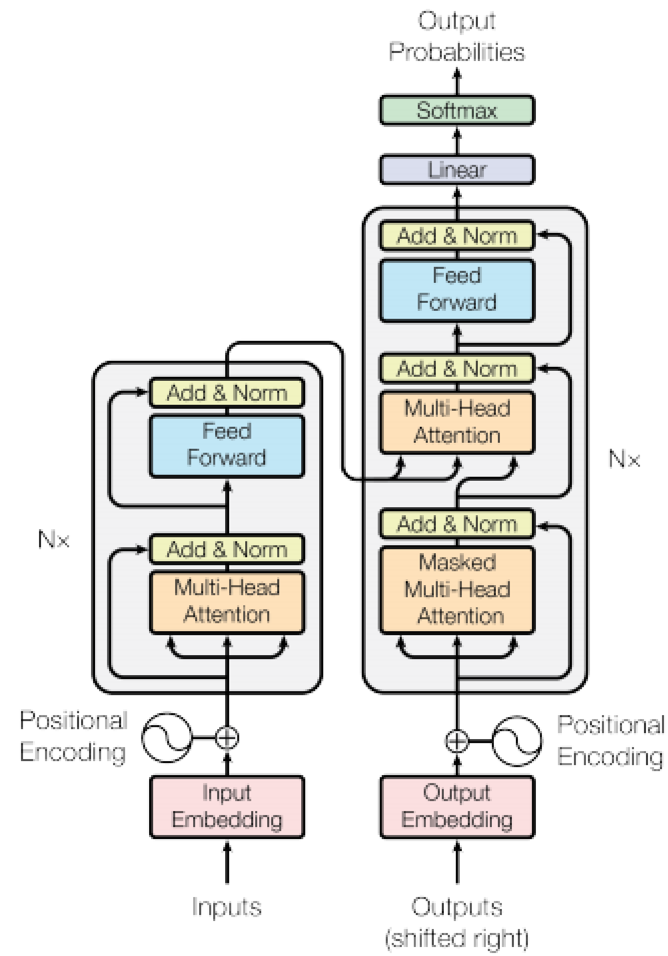
New developments:
Attention is All You Need

RNN: compress history into the state vector

UniRNN: attention over history!

BiRNN: attention over whole sequence

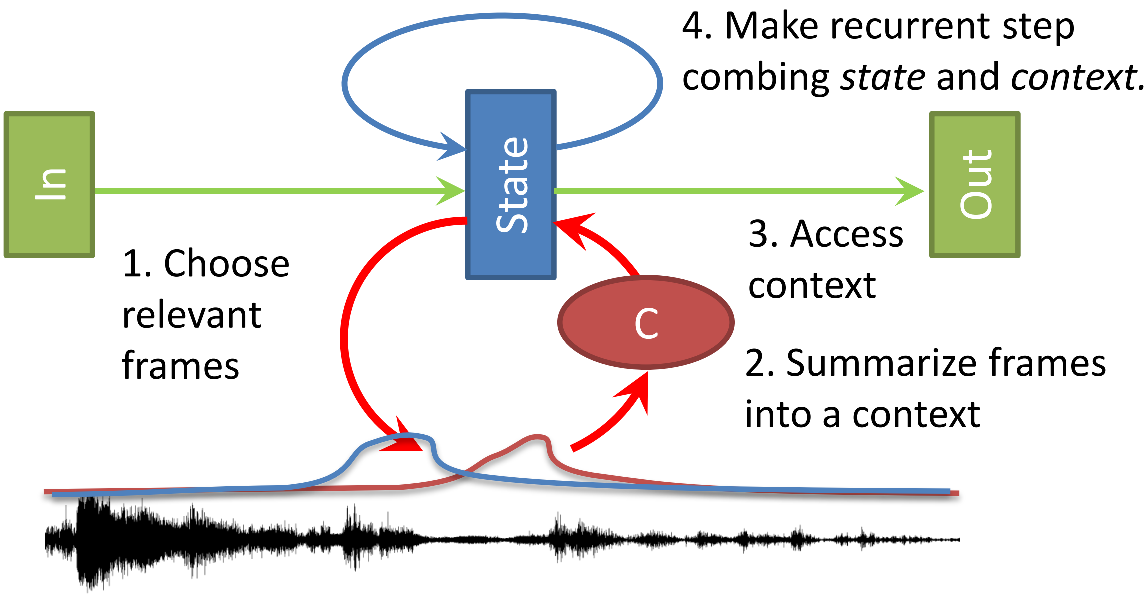




A. Vasvani et al. https://arxiv.org/pdf/1706.03762.pdf

SOME OF MY WORK

Location-aware Attention

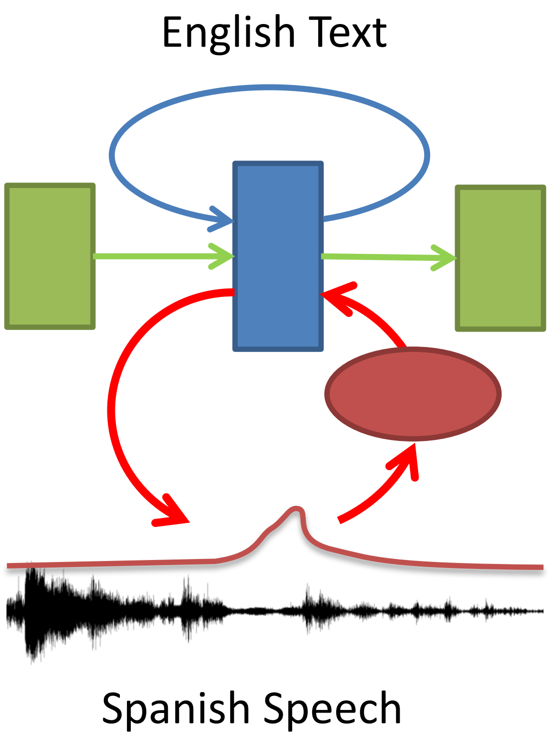


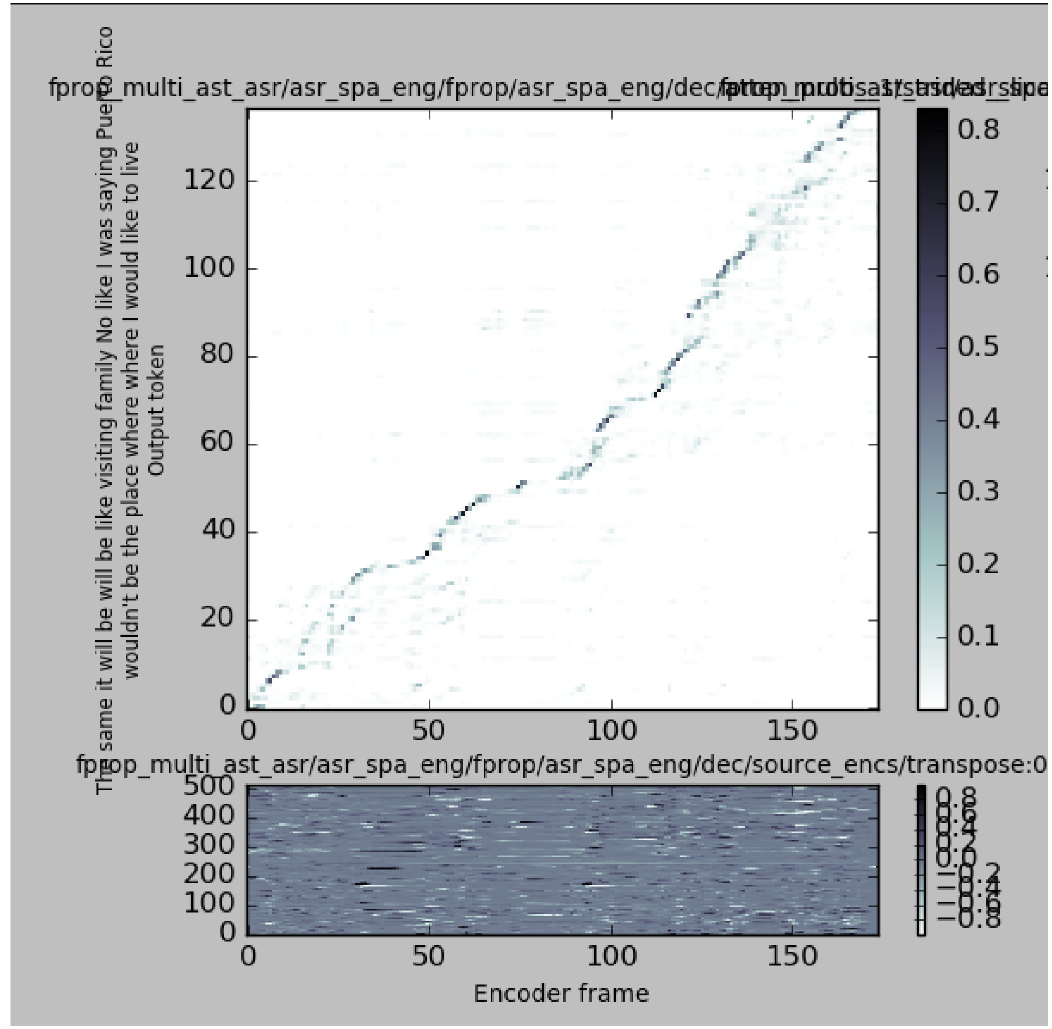
- We want to separate repetitions of the same sound
- Use the selection from the last step to make the new selection
- This enables the model to learn concepts like “later than last” or “close to last”.

Chorowski et al., „Attention-based models for speech recognition”, NIPS 2015

Our approach

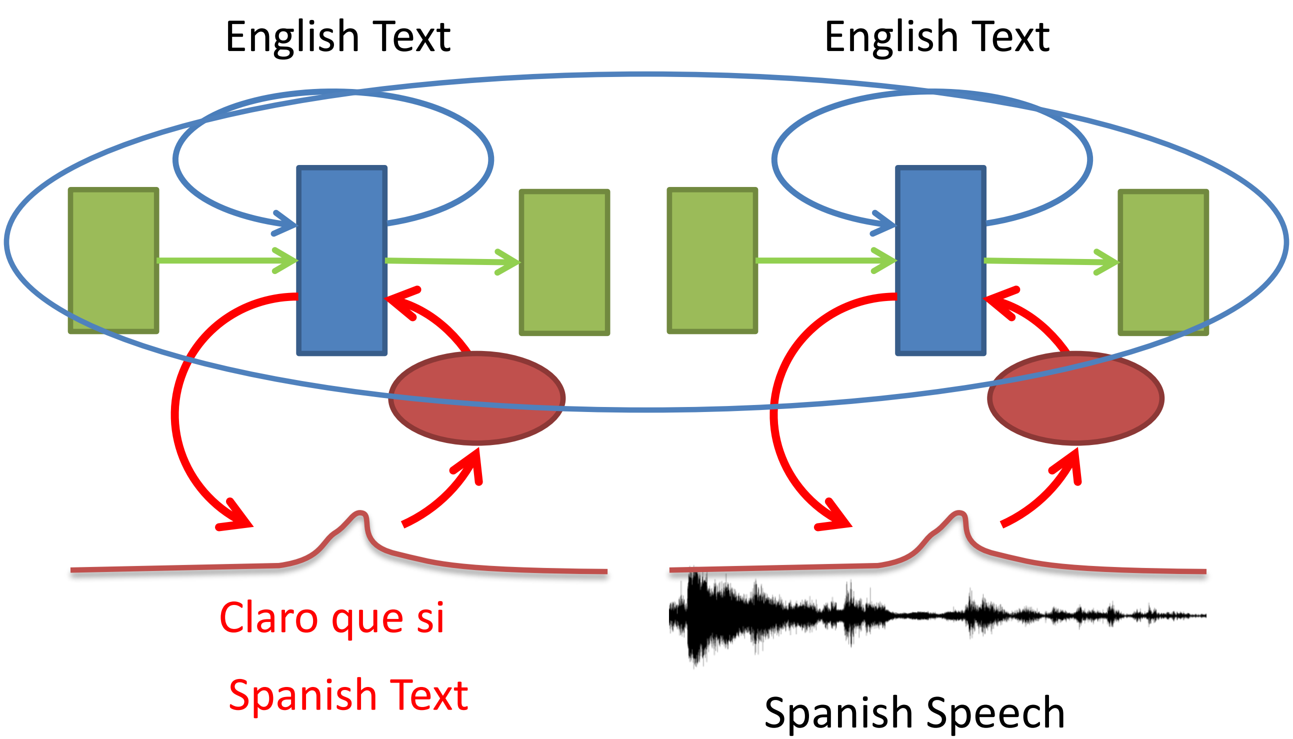
- Seq2seq model





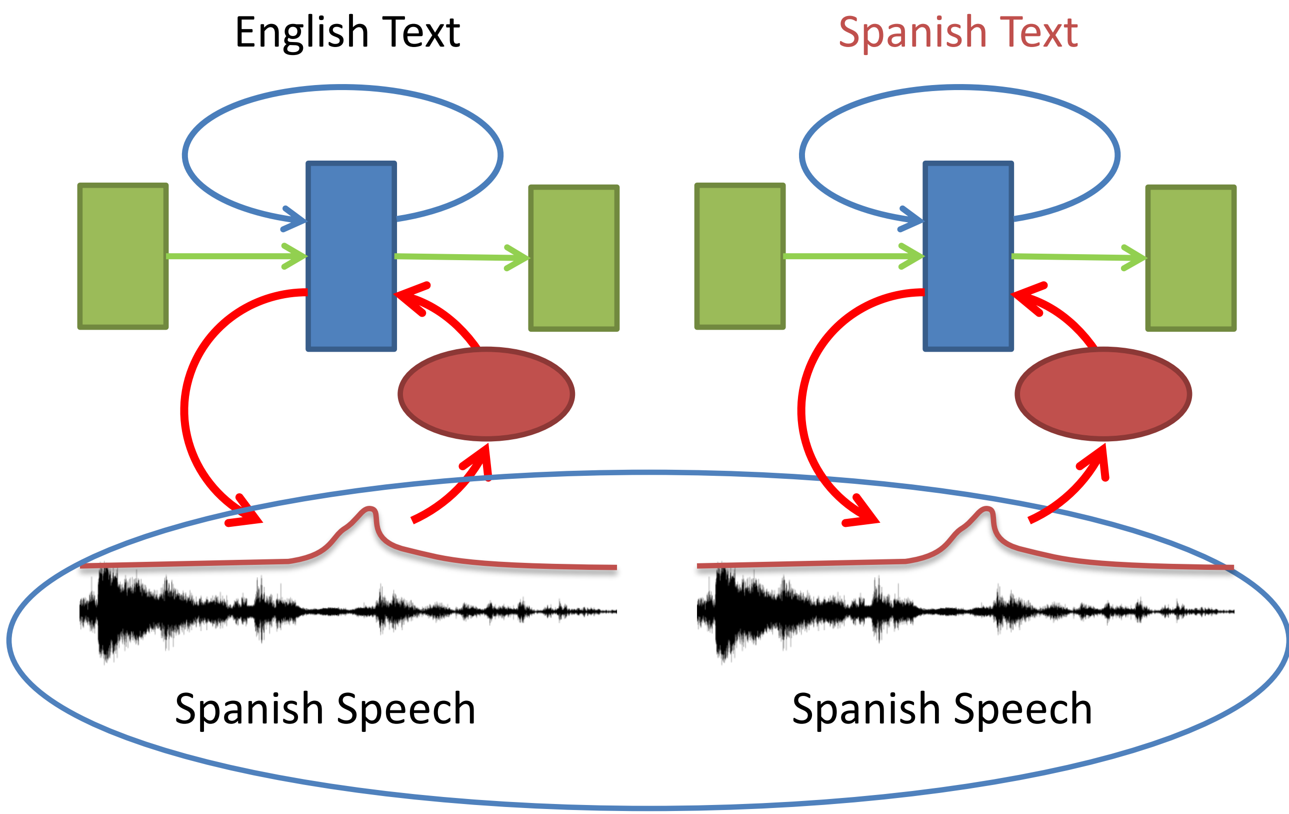
Multitask Learning, or Exploit All Data

Share weights of the decoder, separate encoders

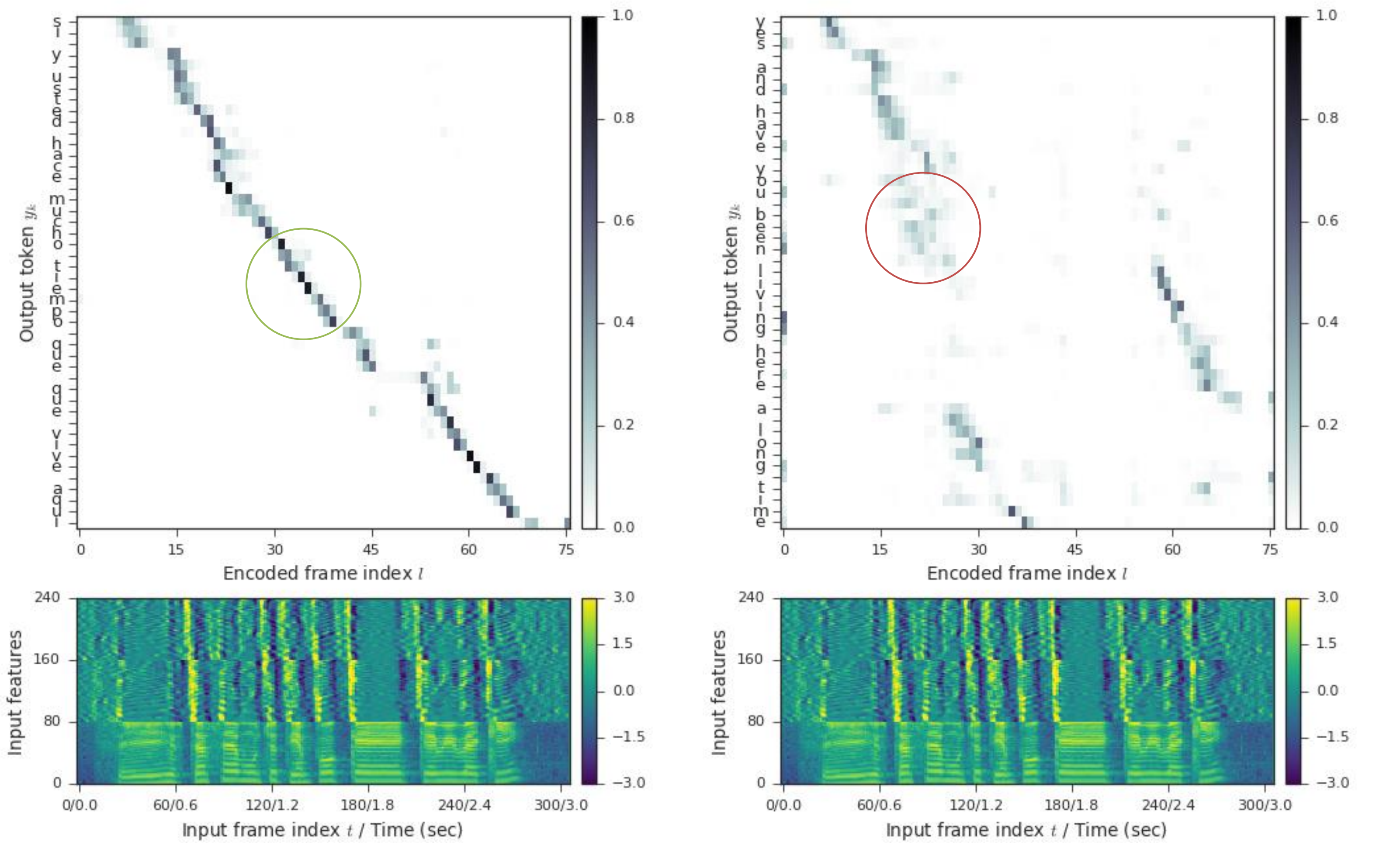


Multitask Learning, or Exploit All Data

Share weights of the encoder, separate decoders

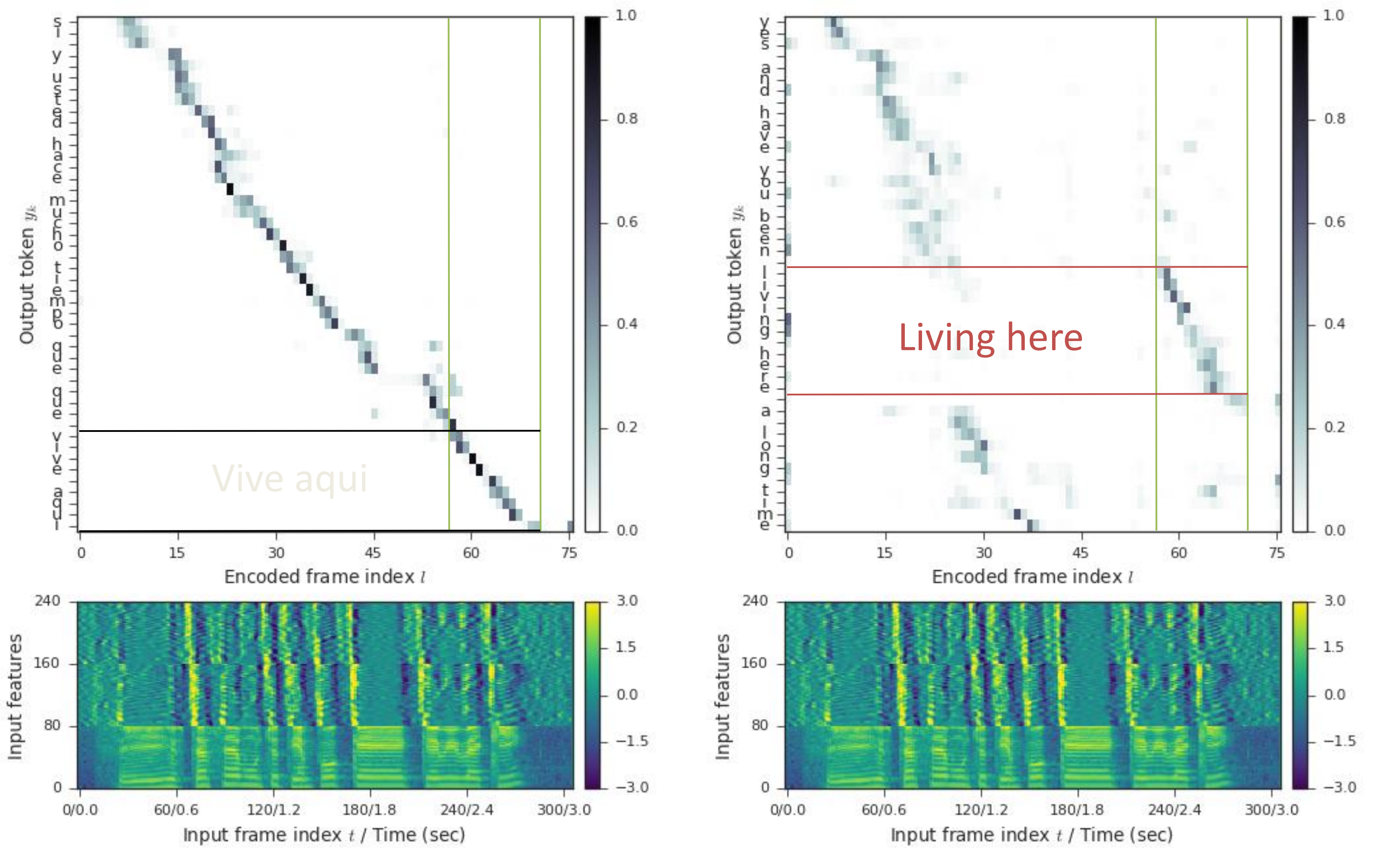


Seq2seq Speech Translation: Attention



- recognition attention very confident
- translation attention smoothed out across many spectrogram frames for each output character
  - ambiguous mapping between Spanish speech acoustics and English text

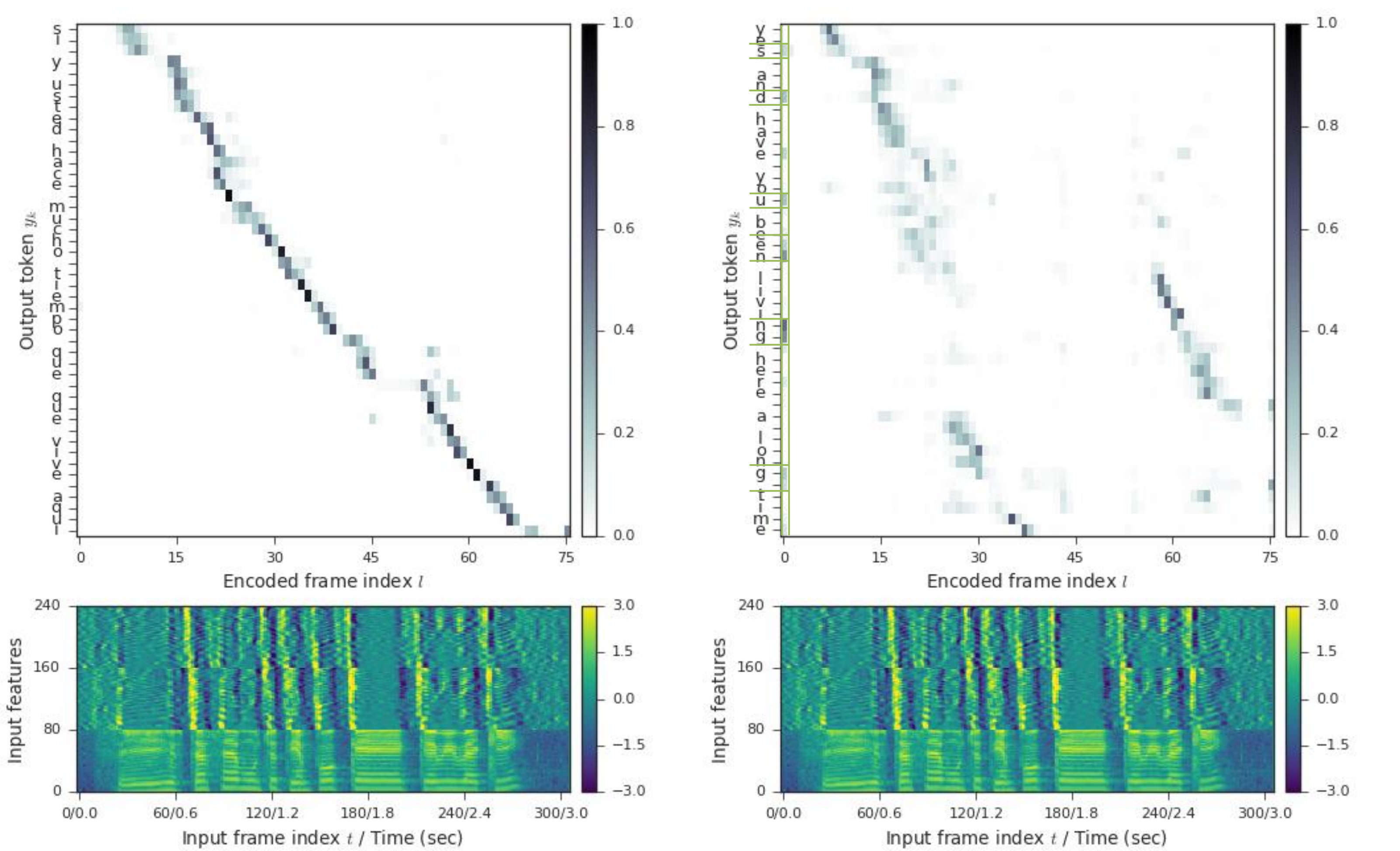
Seq2seq Speech Translation: Attention



- speech recognition attention is mostly monotonic
- translation attention reorders input: same frames attended to for "vive aqui" and "living here"

Weiss, Chorowski et al., Sequence-to-Sequence Models Can Directly Translate Foreign Speech, INTERSPEECH 2017

Seq2seq Speech Translation: Example attention



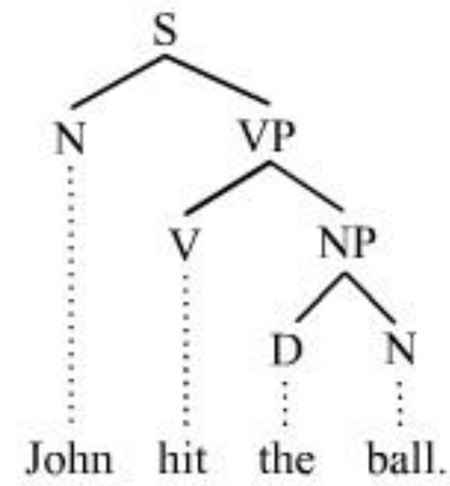
translation model attends to the beginning of input (i.e. silence) for the last few letters in each word

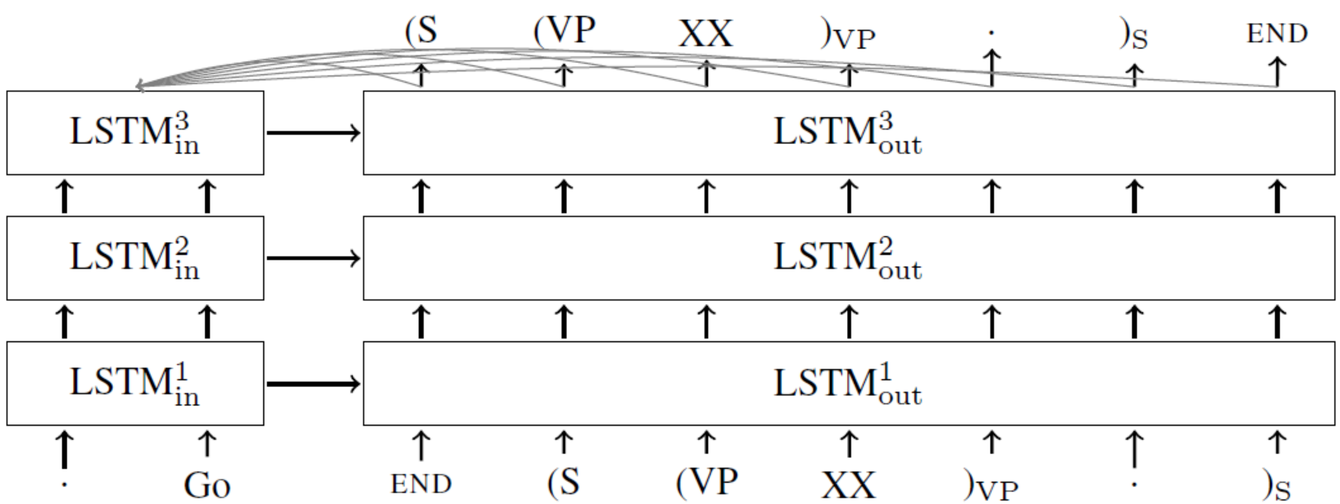
- already made a decision about word to emit, just acts a language model to spell it out.

End-to-end systems in NLP: How to parse sentences?

For constituency parsing:
Treat parsing as a sequence-to-sequence problem:

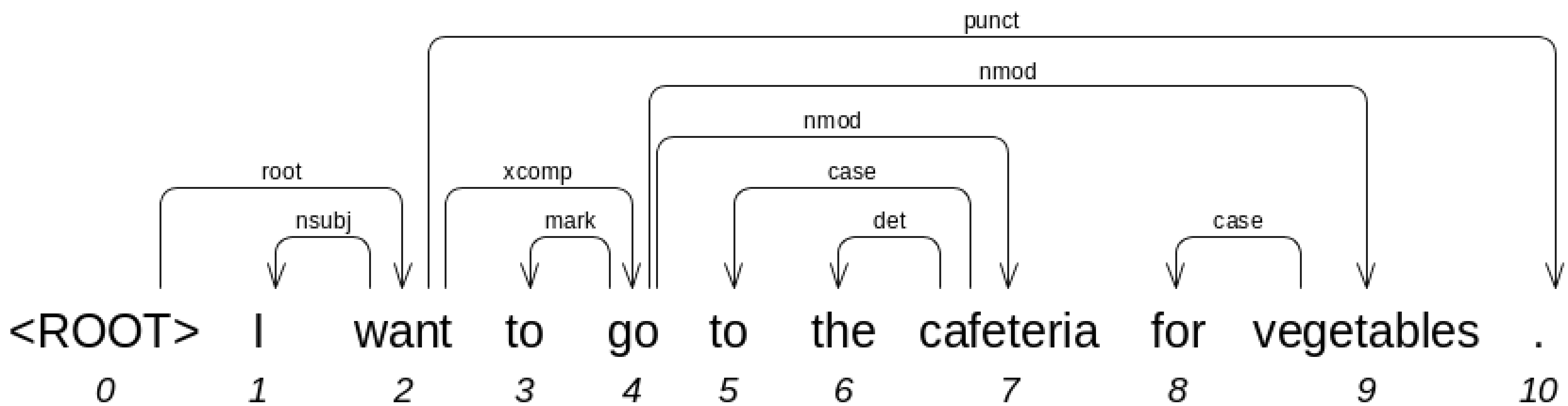
- Input: sentence
  „Go ."
- Output: linearized parse tree:
  „(S (VP XX)VP .)S END"





O. Vinyals et al, "Grammar as a Foreign Language", NIPS 2015

Dependency parsing

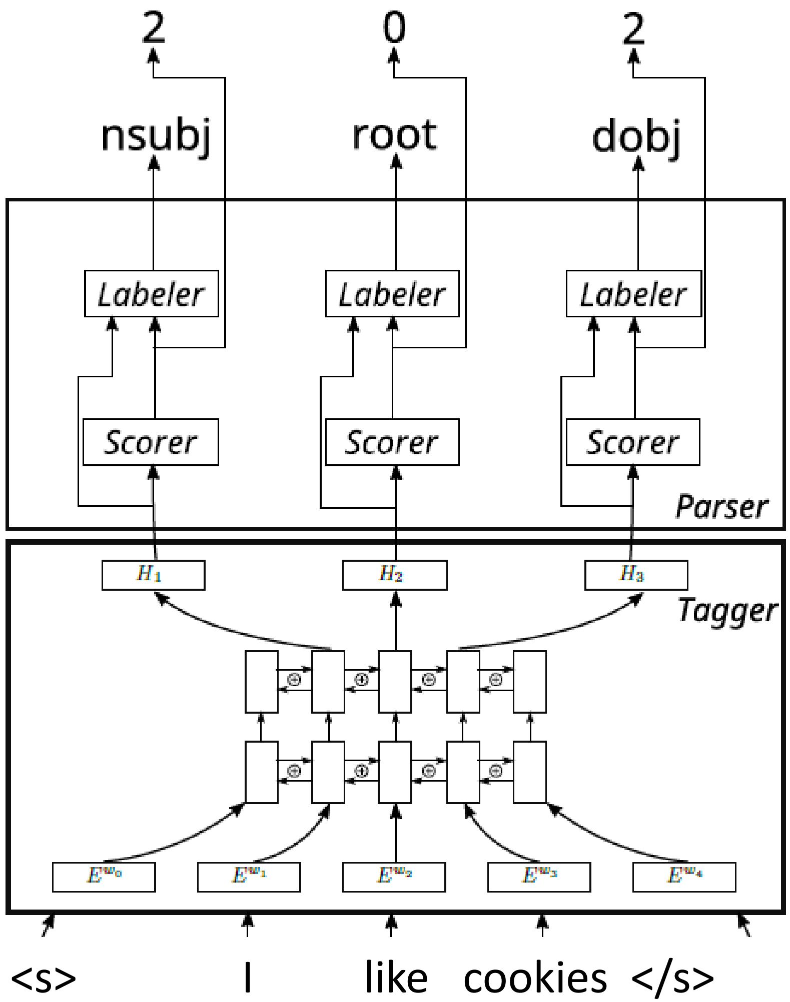


- Desired output: directed edges between words.
- At each step the attention selects a few words.
- Idea: use the selection weights as pointers.

Chorowski et al. “Read, Tag, and Parse All at Once, or Fully-neural Dependency Parsing”, arxiv https://arxiv.org/pdf/1609.03441

Zapotoczny et al. “On Multilingual Training of Neural Dependency Parsers” TSD 2017

Dependency parsing

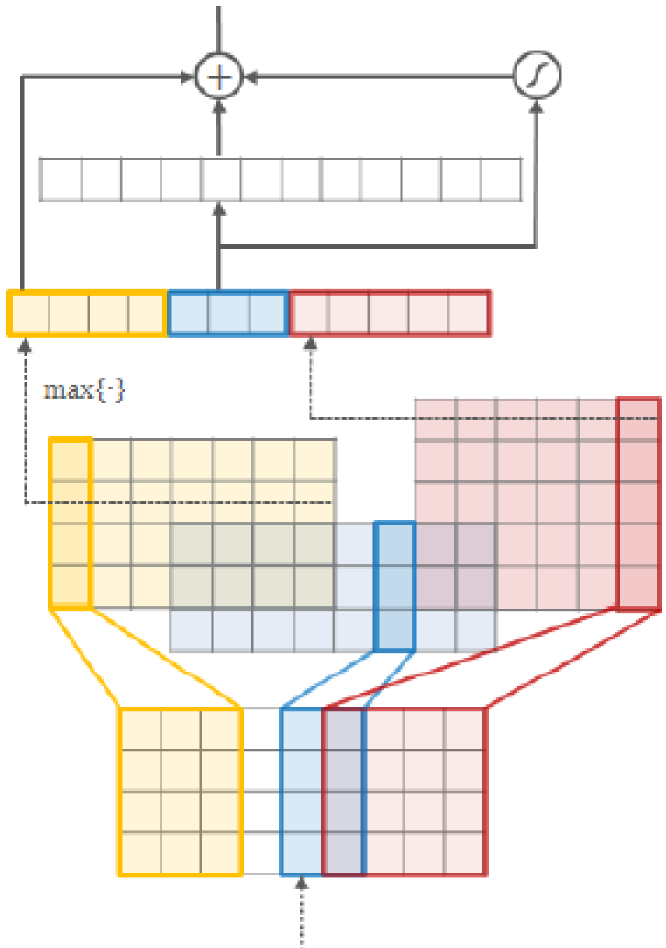


For each word w

Two operations:

1. Find head h (use attention mechanism)
2. Use (w, h) to predict dependency type

From characters to word embeddings



Highway layers – very nonlinear transformation of data

Glue best word representations together

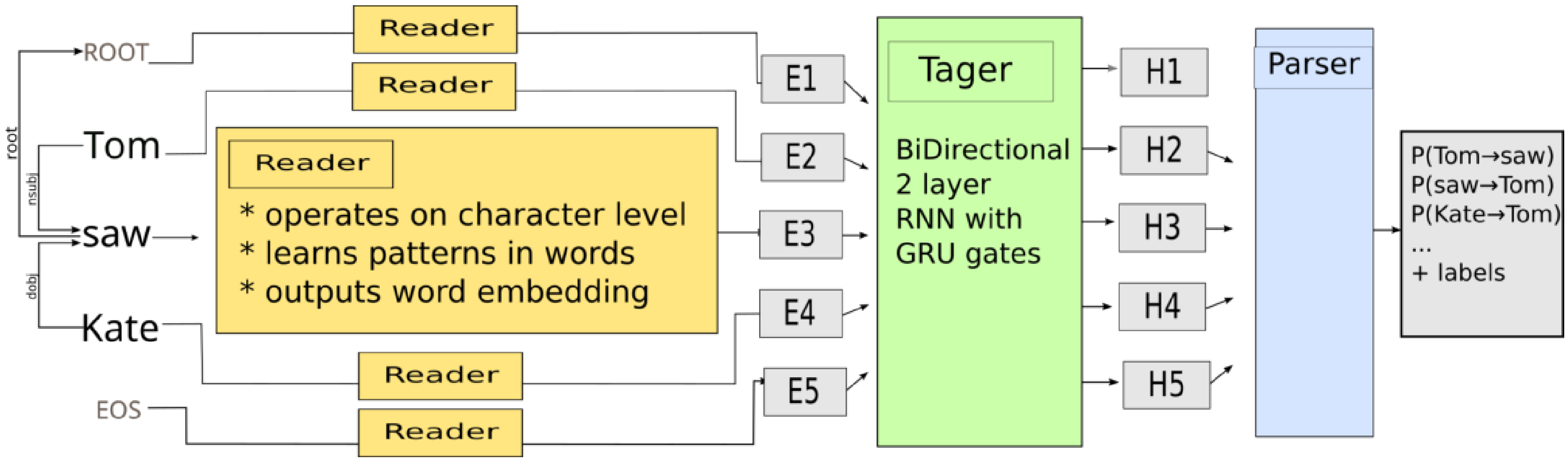
Convolutional filters of varying lengths. Can react to pre-, in-, and post-fixes of words

Character embeddings concatenated into a matrix

„incomprehensible"

Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-Aware Neural Language Models," *arXiv:1508.06615 [cs, stat]*, Aug. 2015.

From characters to parse trees



Reader reads orthographic representations of words and is sensitive to morphemes.

Tagger puts words into context

Parser finds the dependency edges.

Jabberwocky (Lewis Carroll)

Twas brillig and the slithy toves
Did gyre and gimble in the wabe;
All mimsy were the borogoves,
And the mome raths outgrabe.

Żabrołak (Stanisław Barańczak)

Brzdęśniało	już	ślimonne	prztowie
praet:sg:n:perf	qub	adj:sg:nom:n:pos	subst:sg:nom:n

Wyrło	i	warło	się	w	gulbieży
praet:sg:n:perf	conj	praet:sg:n:imperf	qub	prep:acc:nwok	subst:pl:acc:m3

Zmimszałe	ćwiły	borogowie
adj:pl:acc:m3:pos	praet:pl:f:imperf	subst:pl:nom:m1

I	rcie	grdypały	z	mrzerzy
conj	subst:pl:nom:n	praet:pl:f:imperf	prep:gen:nwok	subst:sg:gen:f

Underlined words are neologisms, green are correct!

Multilingual Grammatical Relations

Polish word	Closest russian embedings
przedwrześniowej	адренергической тренерской таврической непосредственной археологической философской *верхнюю*
większych	автомобильных *трёхдневные* технических практических официальных оригинальных
policyjnym	главным историческим глазным непосре-дственным *косыми* летним двухсимвольным

- Green Russian words have similar grammatical function to Polish words.
- -ской (skoy) and -нной (nnoy) quite distant from polish –owej (ovey).
- 3-letter -ych paired with 2 letter -ых