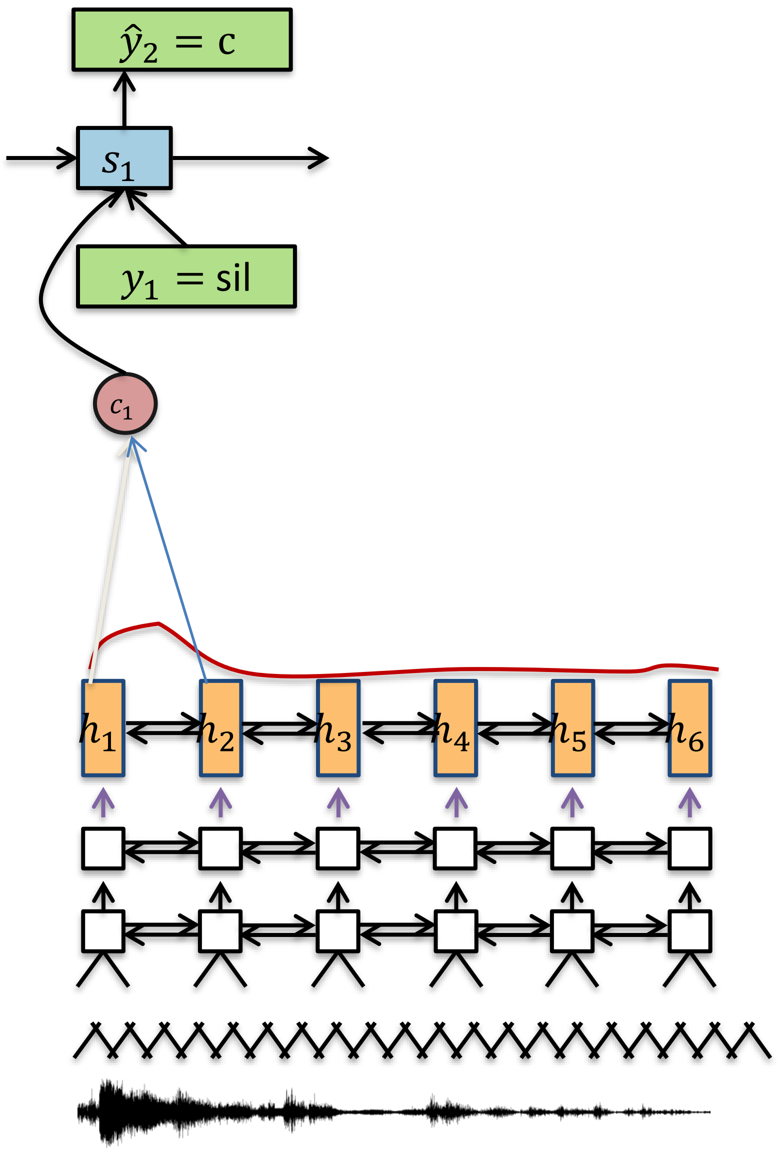
USING ATTENTION FOR SPEECH RECOGNITION AND NLP

Attention ASR at a Glance



"Language model"
RNN Generates text
letter-by-letter

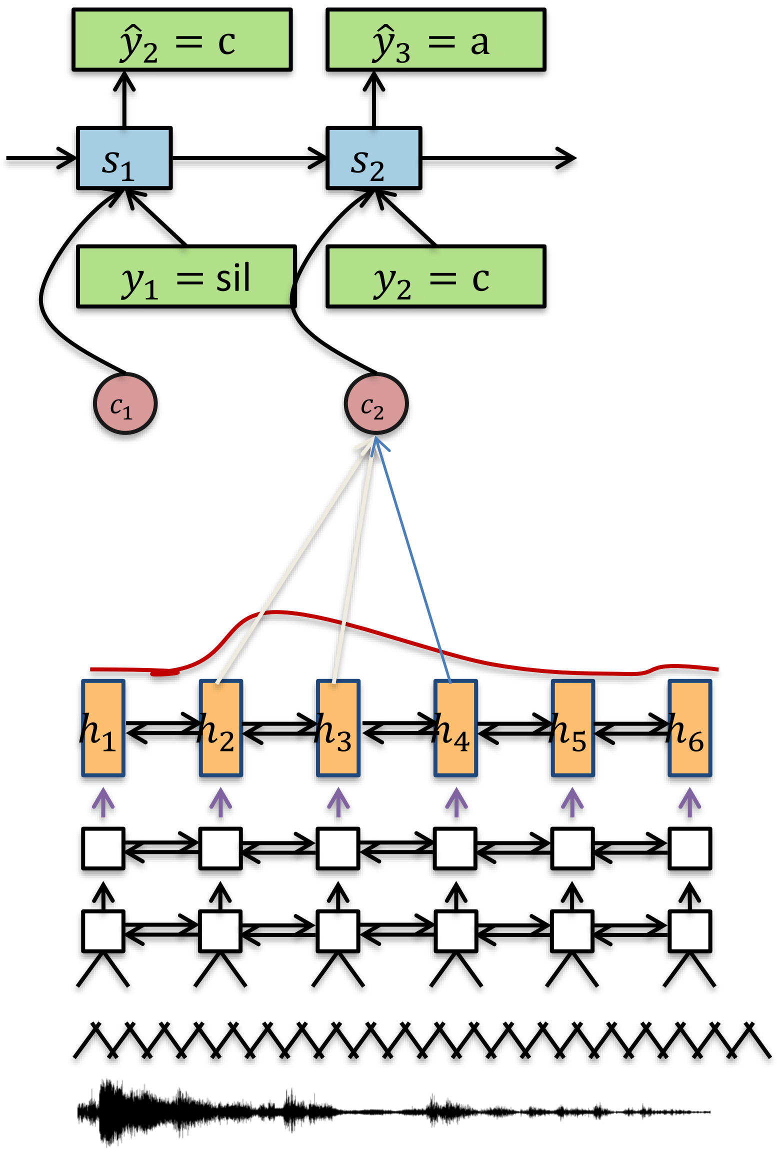
Alignment model:
Attention mechanism

"Acoustic model"
Convolutional and
recurrent layers

Speech features
Mel spectrogram

Chorowski J., Bahdanau, Serdyuk, D., D., Cho, K. Bengio, Y., Attention-Based Models for Speech Recognition, NIPS 2015
Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., Bengio, Y., End-to-End Attention-based Large Vocabulary Speech Recognition, ICASSP 2016

Attention ASR at a Glance



"Language model"
RNN Generates text
letter-by-letter

Alignment model:
Attention mechanism

"Acoustic model"
Convolutional and
recurrent layers

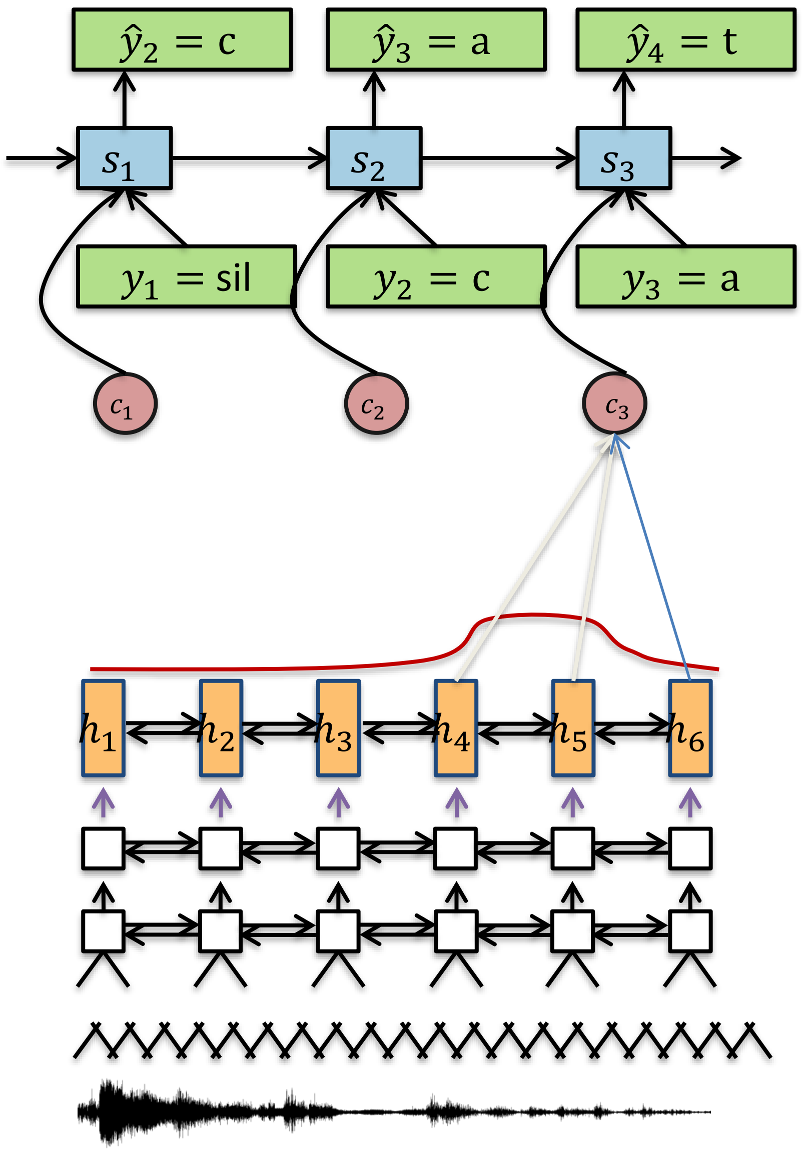
Speech features
Mel spectrogram

Chorowski J., Bahdanau, Serdyuk, D., D., Cho, K. Bengio, Y., Attention-Based Models for Speech Recognition, NIPS 2015
Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., Bengio, Y., End-to-End Attention-based Large Vocabulary Speech Recognition, ICASSP 2016

Attention ASR at a Glance

Network defines
$p(\text{Words}|\text{Audio};\Theta)$
where
Θ are parameters.

Training uses
gradient
optimization



"Language model"
RNN Generates text
letter-by-letter

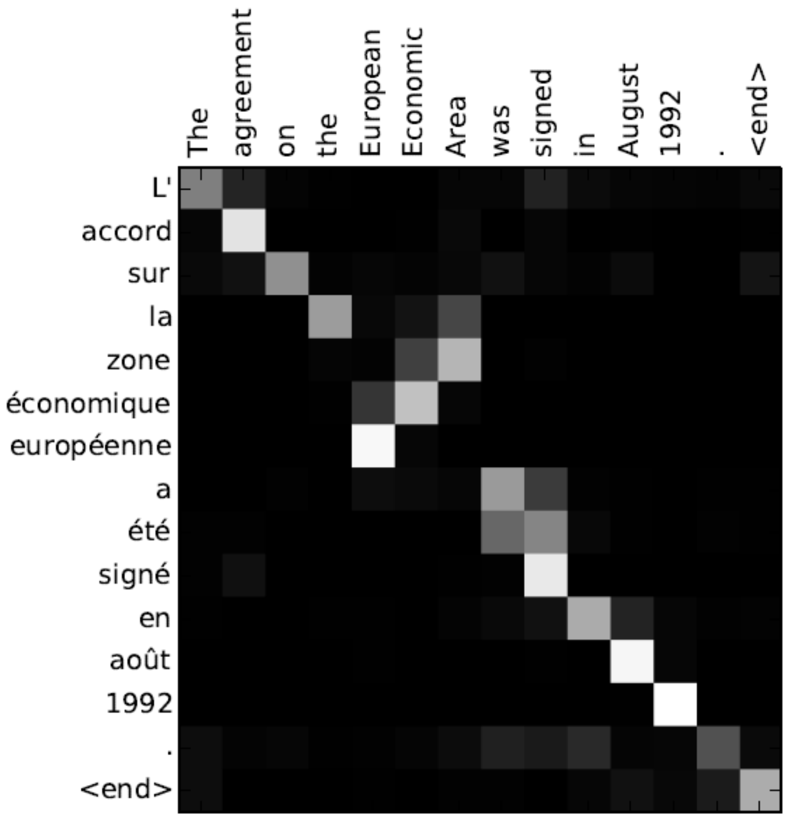
Alignment model:
Attention mechanism

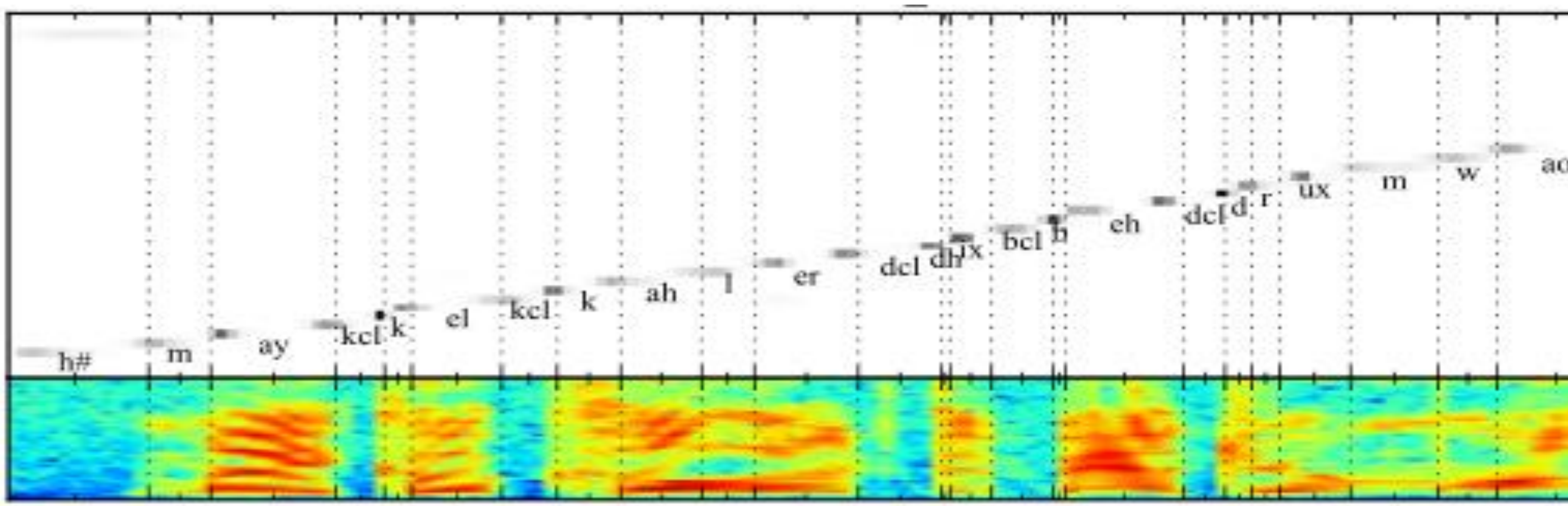
"Acoustic model"
Convolutional and
recurrent layers

Speech features
Mel spectrogram

Chorowski J., Bahdanau, Serdyuk, D., D., Cho, K. Bengio, Y., Attention-Based Models for Speech Recognition, NIPS 2015
Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., Bengio, Y., End-to-End Attention-based Large Vocabulary Speech Recognition, ICASSP 2016

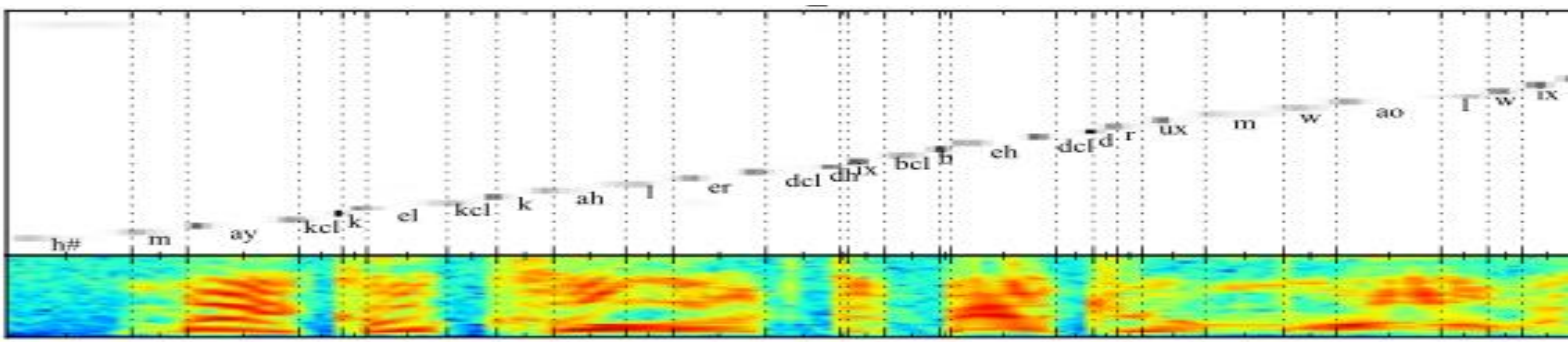
Attention Mechanism in Action





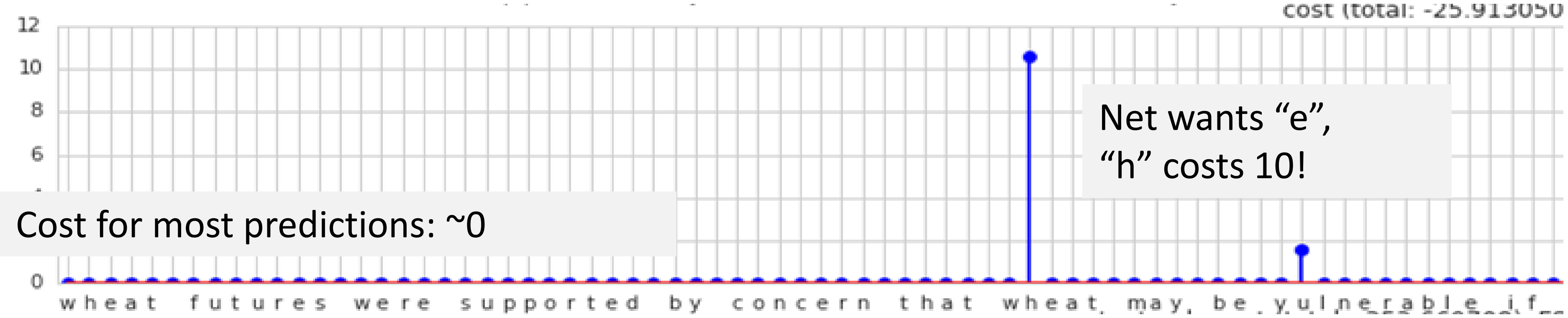
Challenges

- Overconfidence.
- Long sequences and repetitions.
- Language model integration and coverage.

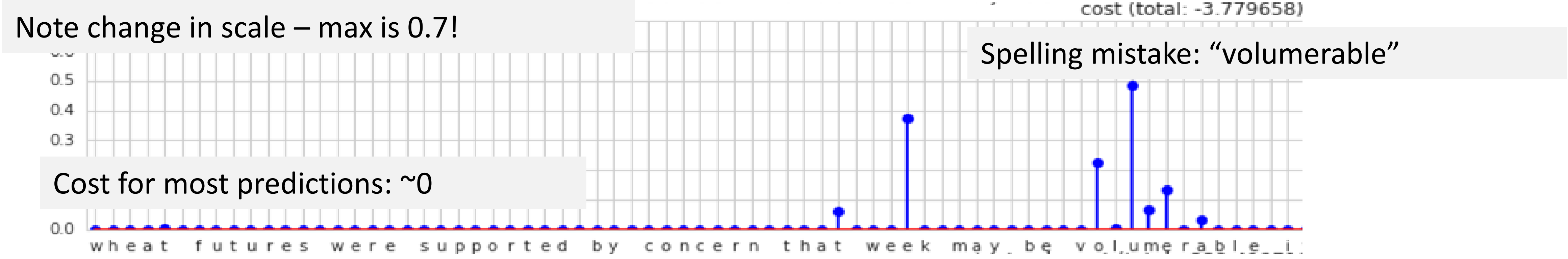


Overconfidence

Ground truth, total log probability -25



Beam search result: total log probability -3.7



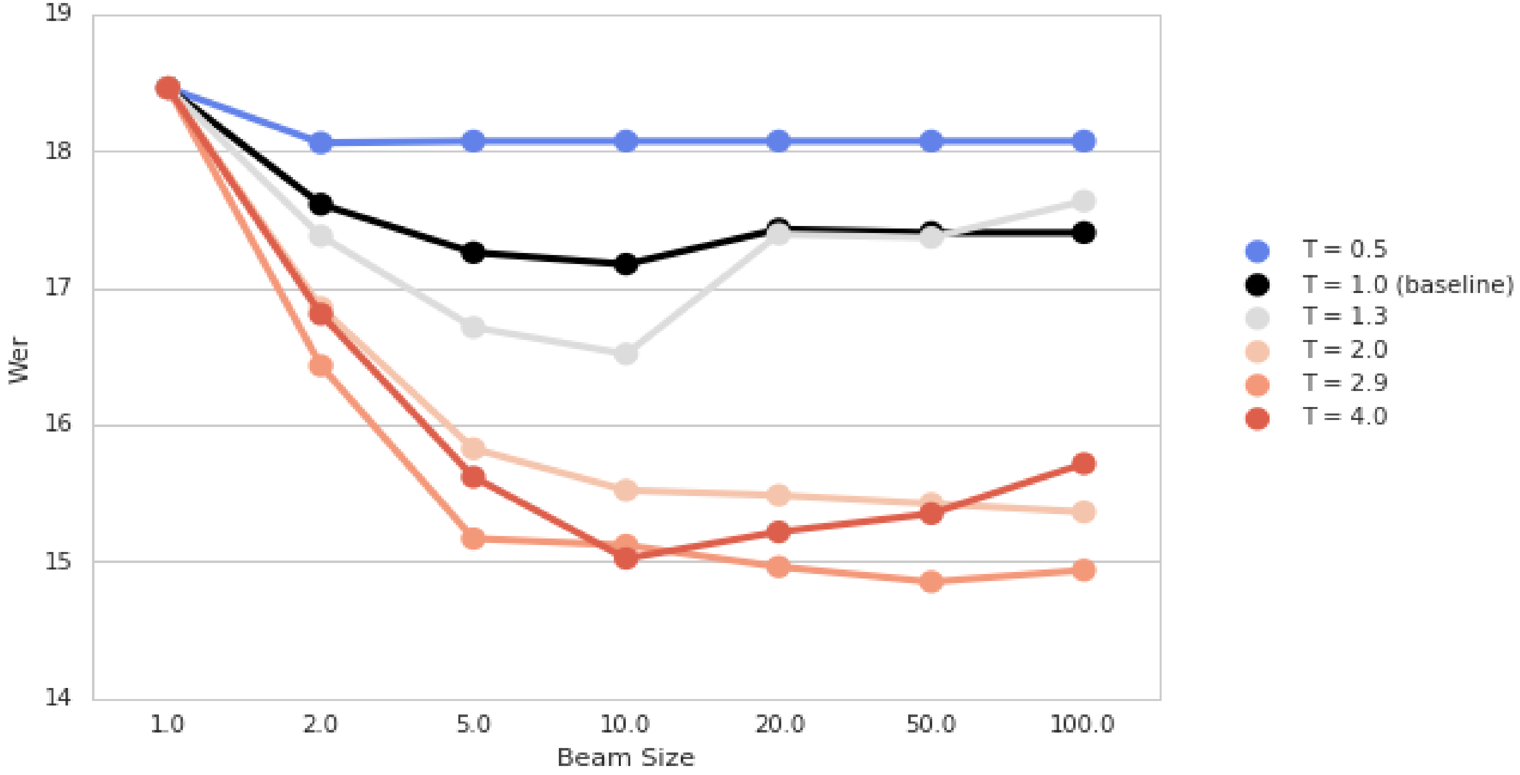
Key Observations

- Accurate next-step predictions:
  99.9% train/96% test
- Overconfidence:
  p(first guess) >> p(second guess)
- A “second guess” of the net costs as much as several “first guess” predictions
  - Beam search ineffective at large beams
  - Very hard to balance decoding costs (e.g. LM)

A Simple Experiment

- After training, tweak SoftMax temperature

$$\text{SoftMax}(Y) = \frac{\exp(Y_i/T)}{\sum_j \exp(Y_j/T)}$$



Training With 1-hot Labels

- The cross-entropy cost for one utterance

$$-\sum_{i=1}^{N}\sum_{c}[\mathrm{Y_i} = \mathrm{c}]\log p_{\Theta}(Y_i|Y_{<i}, X_i)$$

- When model is 99% accurate…
- The only way to reduce cost is to make $p_{\Theta}(Y_i|Y_{<i}, X_i)$ a Dirac delta…

Training With Label Smoothing

- Introduced in Inception V2 (arXiv:1512.00567)

- Change the cost to:

$[\mathrm{Y_i} = \mathrm{c}]$

$$-\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C}\mathrm{T(Y_i, c)}\log p_{\Theta}(Y_i|X_i)$$

- $\mathrm{T}(Y_i, c)$ is a smoothing distribution, e.g.

$$T(Y_i, c) = \begin{cases} \beta, & \text{when } Y_i = c \\ \dfrac{1-\beta}{C-1}, & \text{otherwise} \end{cases}$$

- Even better: smooth the $1-\beta$ according to class marginal probabilities (unigrams)

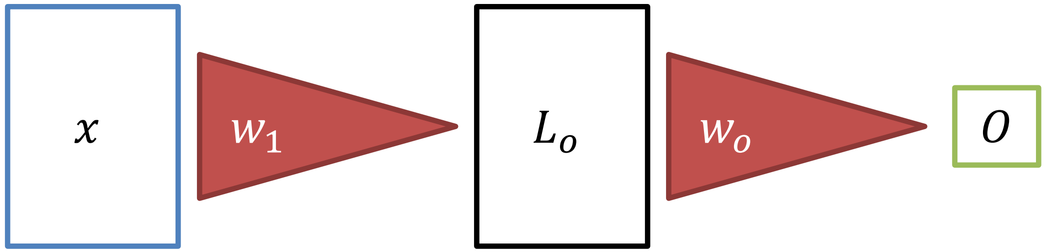
Effects of Label Smoothing

- Reduces overconfidence and regularizes
- Also prevents gradient vanishing:
  - Without smoothing SoftMax derivative is $p_{\Theta}(Y_i|X_i) - [Y_i = c]$
  - This vanishes when $p_{\Theta}(Y_i|X_i) \approx 1$
  - Effectively the model stops training on correctly classified characters

Label Smoothing vs Other Regularizers

At a high level, all regularizers want to forbid large changes of output for small changes of input.

- E.g. weight decay

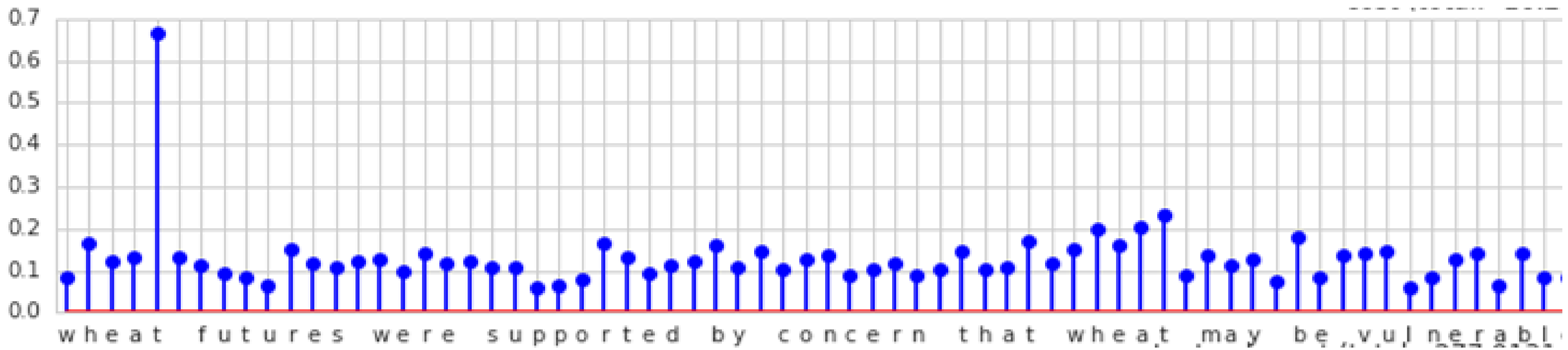


Magnitude of w_o controls the output sensitivity $\frac{\partial O}{\partial L_o} = w_o^T$

- Label smoothing may be easier to use:
  - Easy to say how smooth the output should be
  - Hard to say how large the weights should be

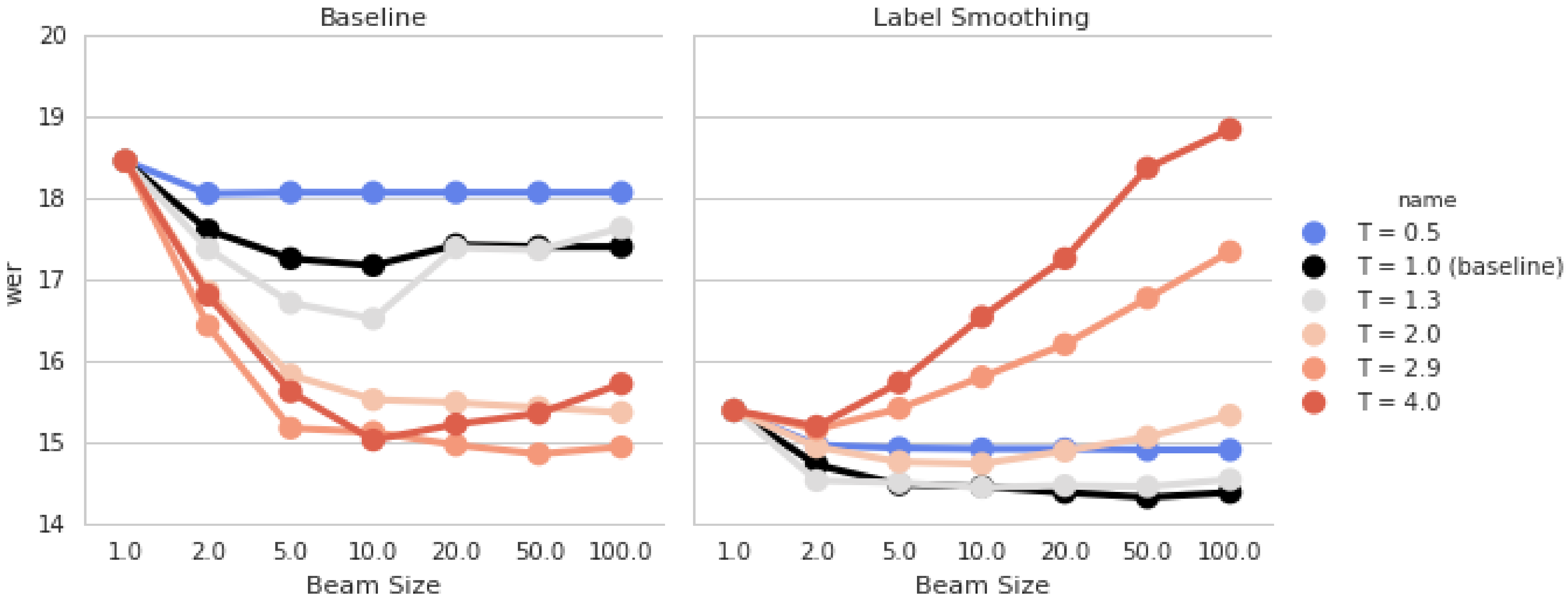
Effects of Label smoothing

- Regularization (next character accuracy increase 96% -> 97%)
- Increase of neg log-probability of best predictions -> other costs easier to balance



SoftMax Temperature and Label Smoothing

- Temperature tweaking no longer needed:



Trouble With Long Sequences

A simple experiment:

1. Train a network as usual.
2. Concatenate test utterances a few times.
3. Decode as usual.

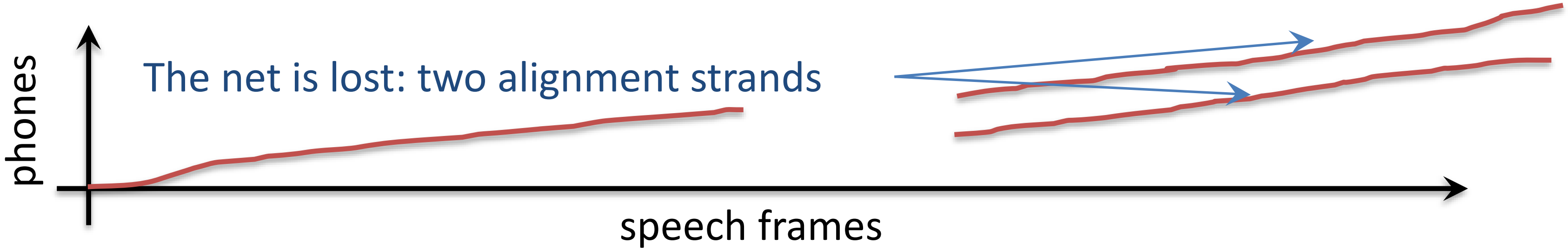
Performance drops dramatically.

On long utterances decoding completely fails.

Investigation of Long Inputs

The setup:
- concatenate utterances
- do force alignment (feed the correct inputs)

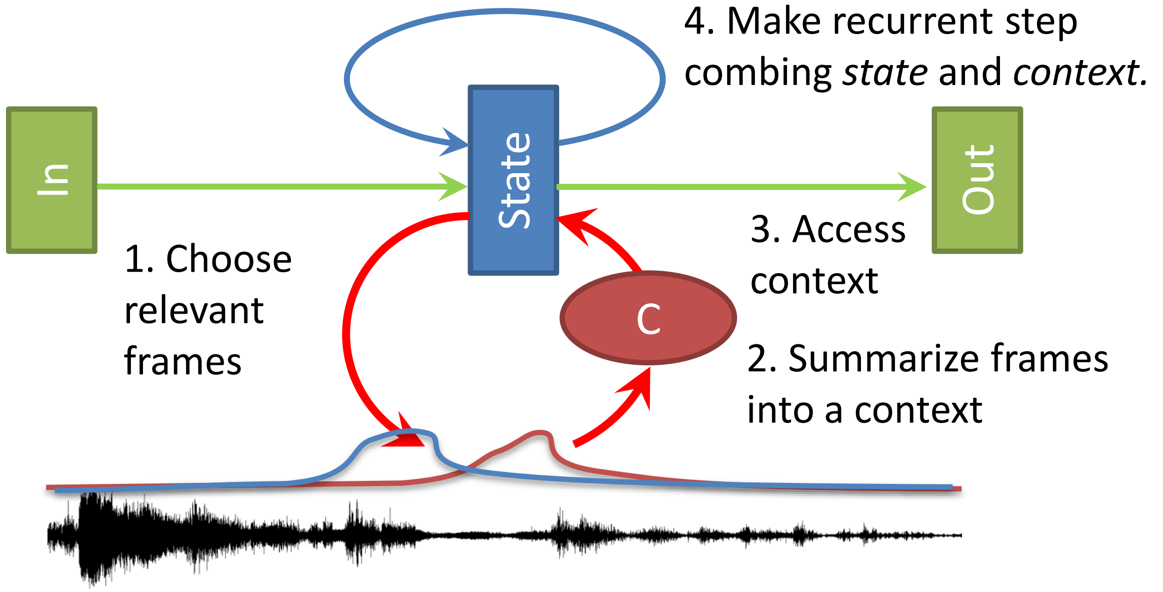
Typical result



Our hypothesis: the net learns an implicit location encoder. It is not robust to long utterances.

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

Location-aware Attention

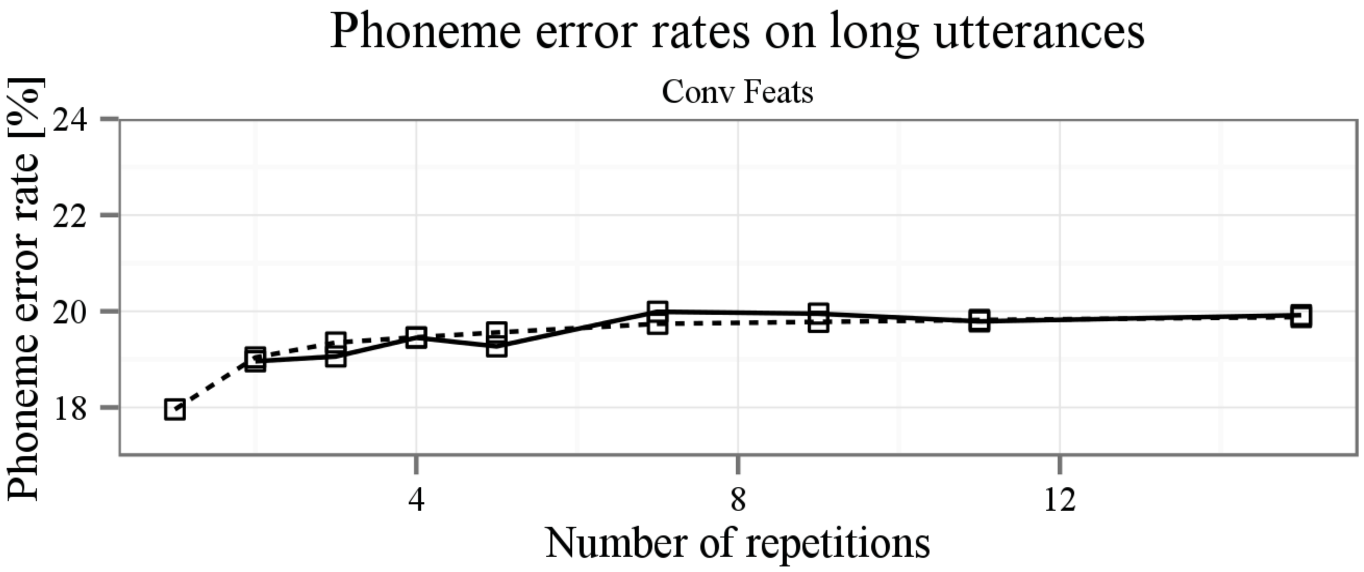


- We want to separate repetitions of the same sound
- Use the selection from the last step to make the new selection
- This enables the model to learn concepts like “later than last” or “close to last”.

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

Location-aware attention helps

- Decoding error rate increases from 18% to 20%



- One more "trick": constrain the attention mechanism to select only few frames
  - Keep up to K with highest scores
  - Limit selection to the vicinity of previous one

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

Decoding With Language Models

- Extend the beam search cost

$$\hat{Y} = \arg\min_{Y} - \log p_{\Theta}(Y|X) - \alpha p_{LM}(Y)$$

Transcript	LM cost $\log p(y)$	Model cost $\log p(y\|x)$	
"chase is nigeria's registrar and the society is an independent organization hired to count votes"	-108.5	-34.5	Ground truth
"in the society is an independent organization hired to count votes"	-64.6	-19.9	Decoded
"chase is nigeria's registrar"	-40.6	-31.2	Severe Transcript Truncation
"chase's nature is register"	-37.8	-20.3	
""	-3.5	-12.5	

Promoting long transcripts

Seems easy:

$$\hat{Y} = \arg\min_{Y} -\log p_{\Theta}(Y|X) - \alpha p_{LM}(Y) - \beta|\mathrm{Y}|$$

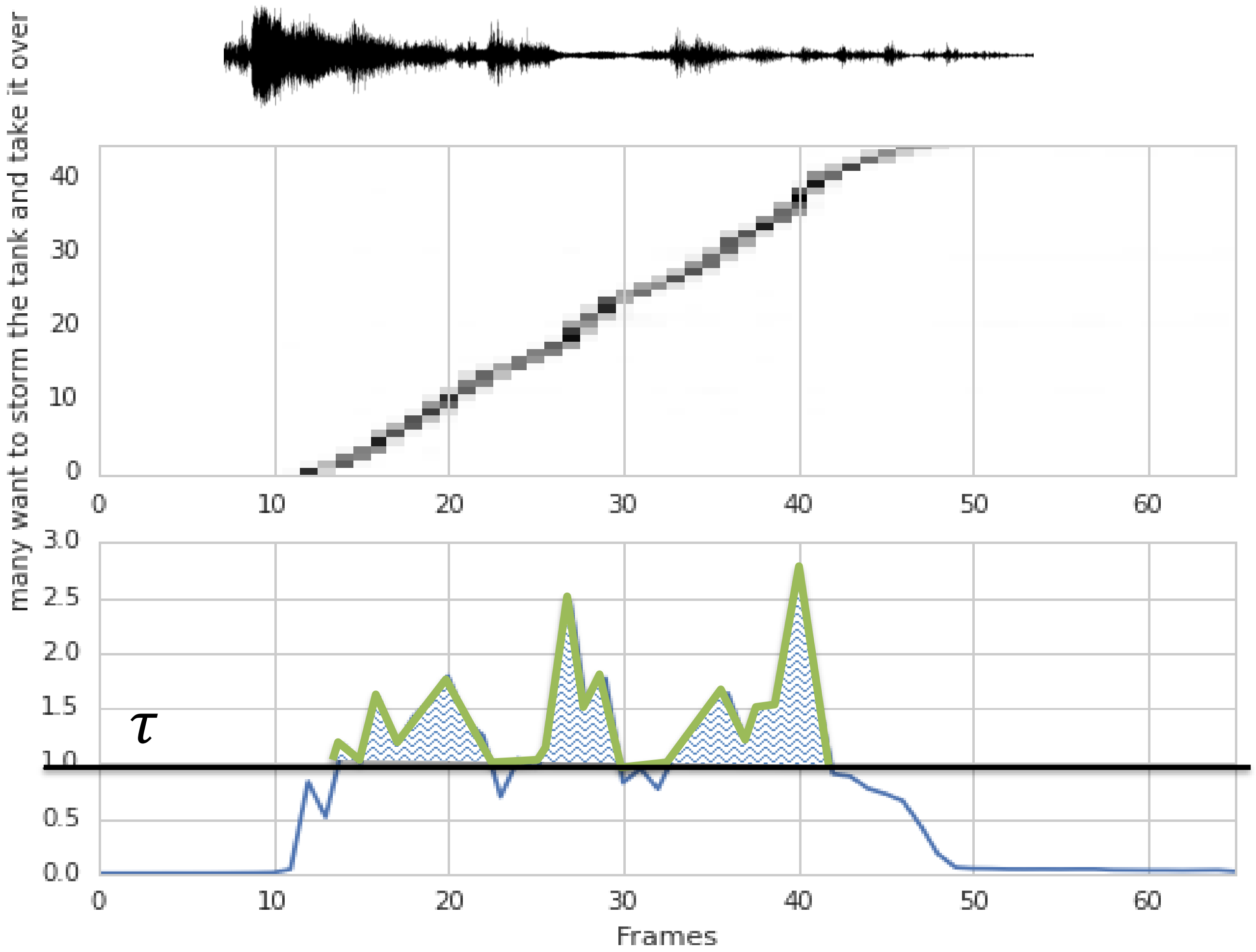
Problem: if any sequence of characters is cheap and the cost becomes negative, the model will keep repeating itself…

Coverage Criterion

Force decoding of all frames, but prevent looping.

$$\text{coverage} = \sum_f [\sum_i \alpha_{fi} > \tau]$$

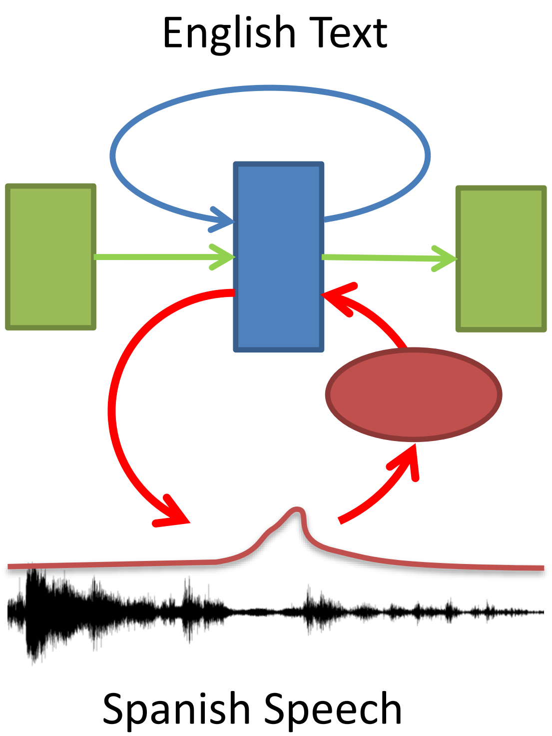
Can't loop: a frame is counted at most once

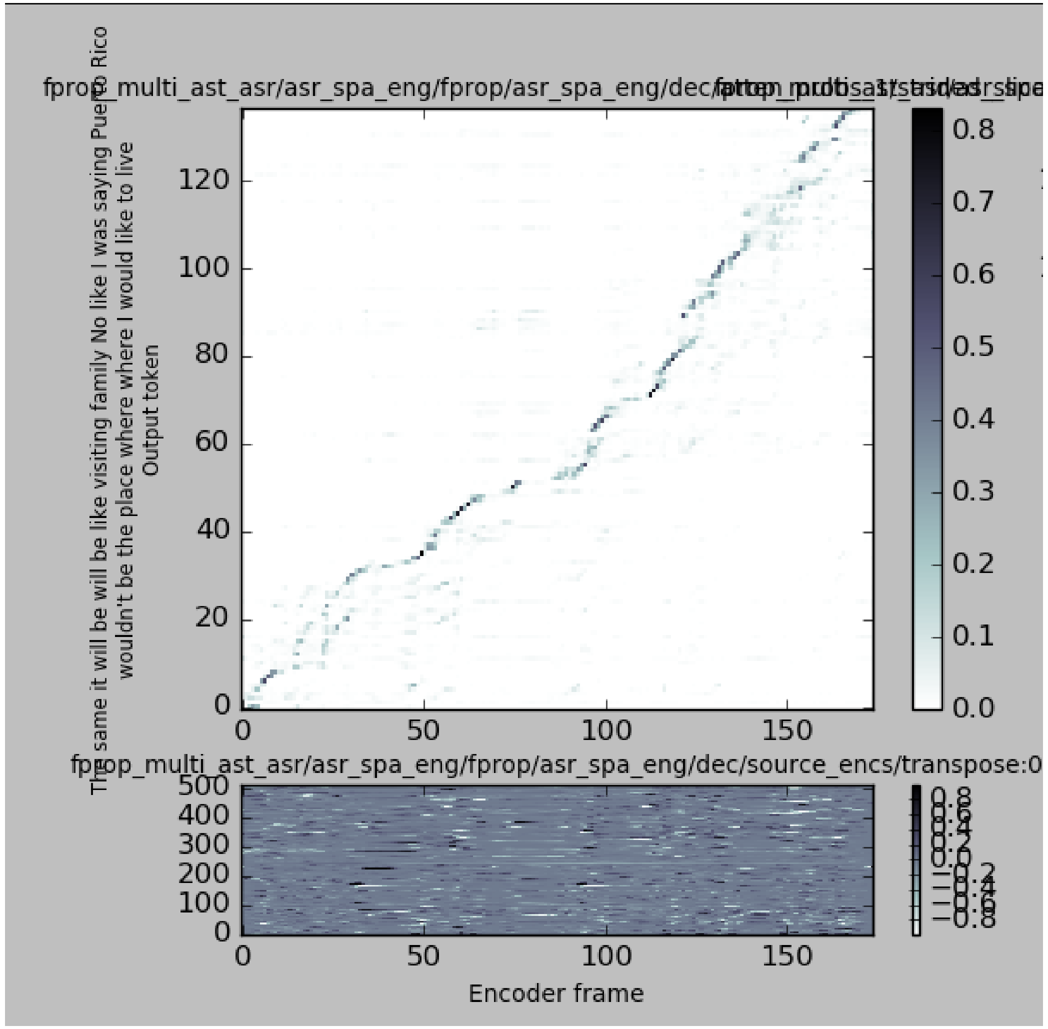


BEYOND SIMPLE SPEECH RECOGNITION

Speech-to-text translation

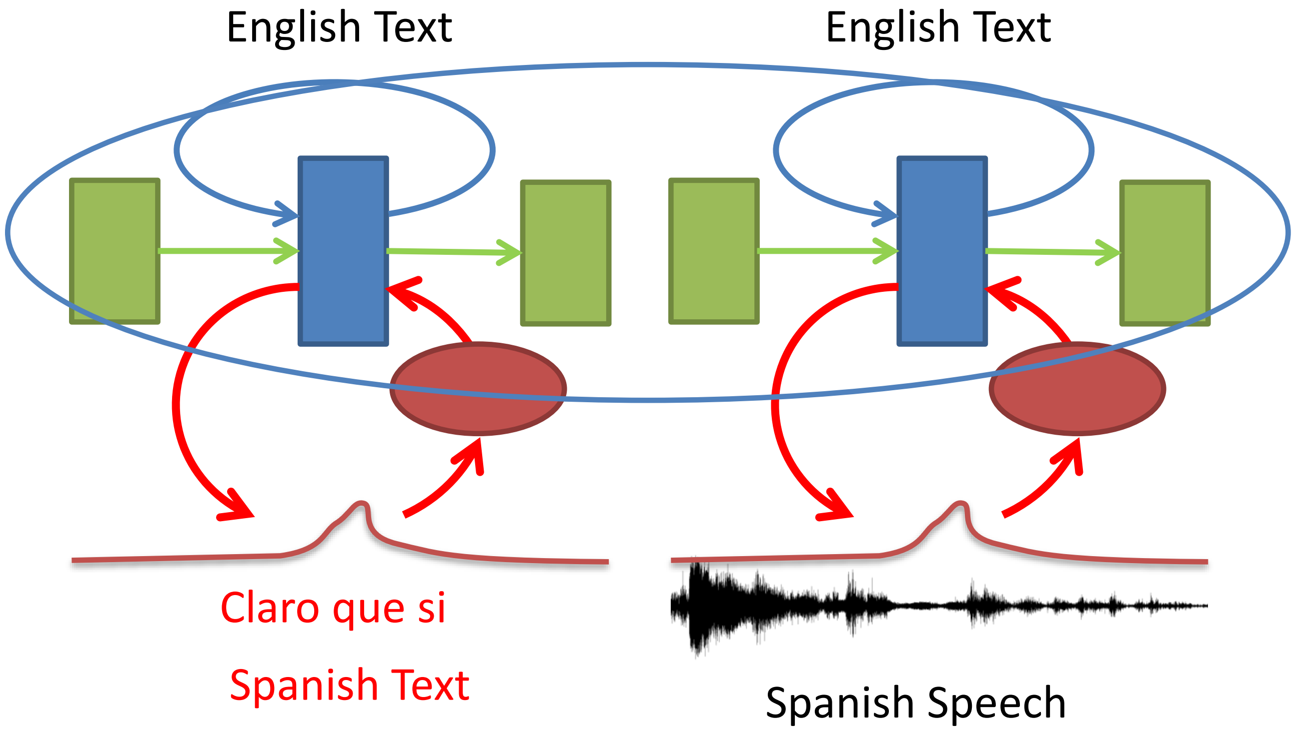
- Seq2seq model





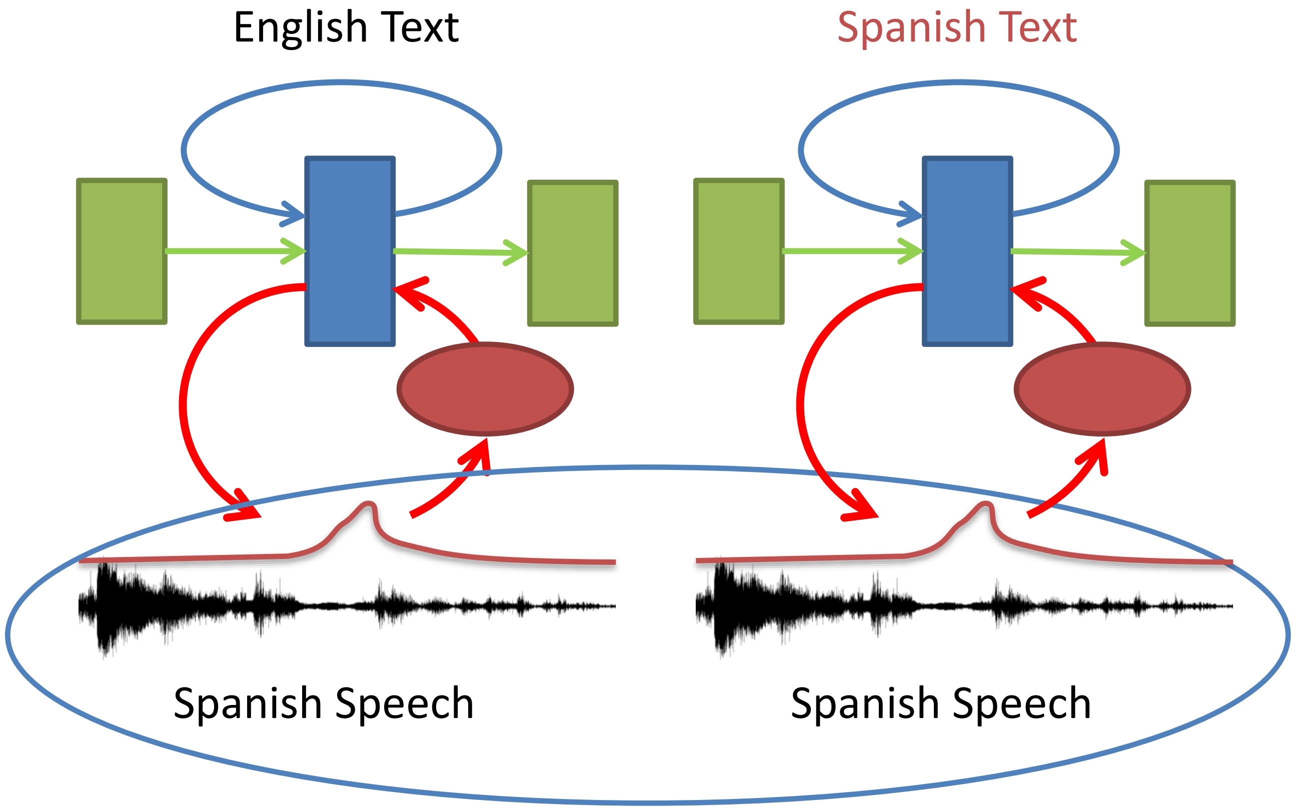
Multitask Learning, or Exploit All Data

Share weights of the decoder, separate encoders

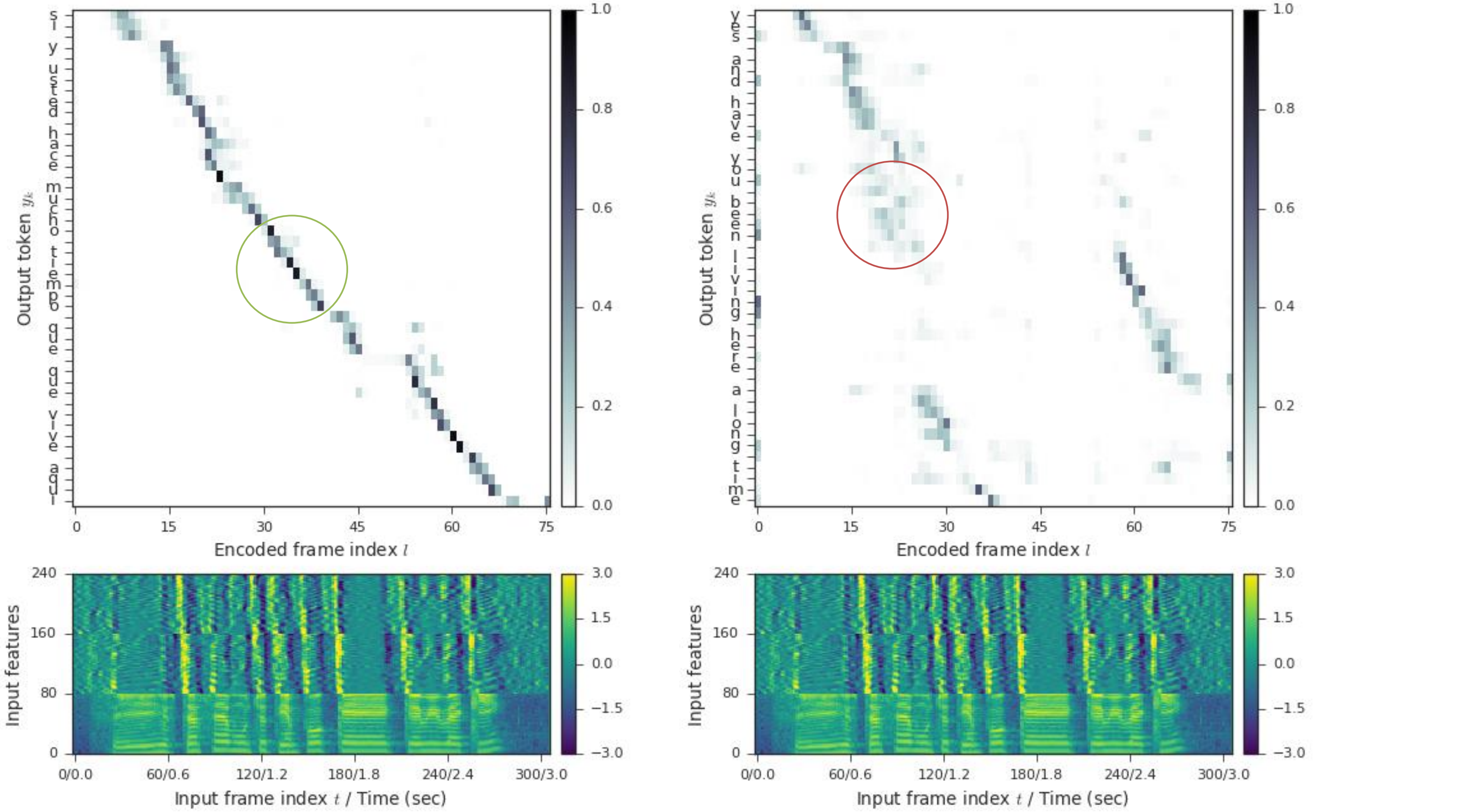


Multitask Learning, or Exploit All Data

Share weights of the encoder, separate decoders

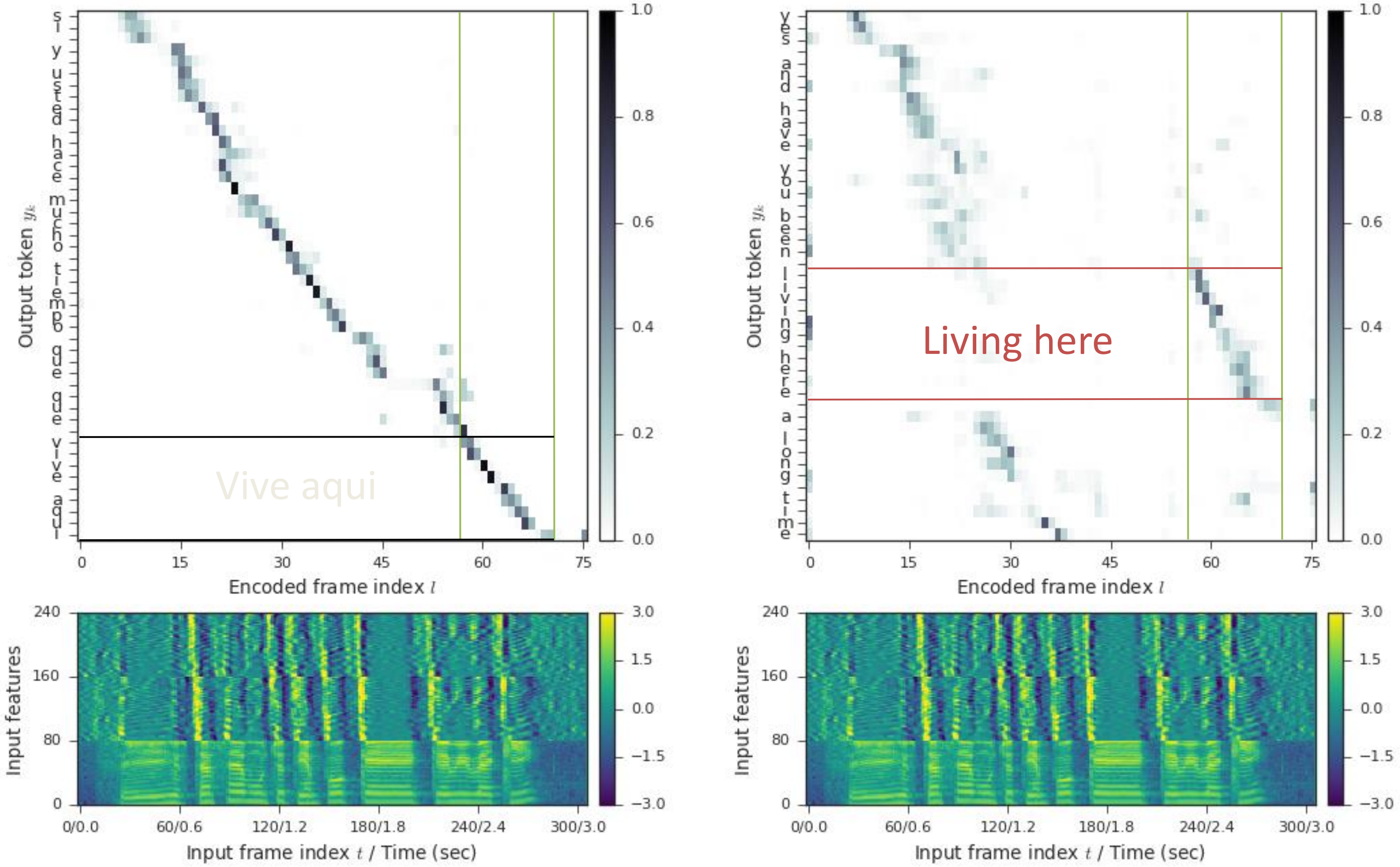


Seq2seq Speech Translation: Attention



- 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

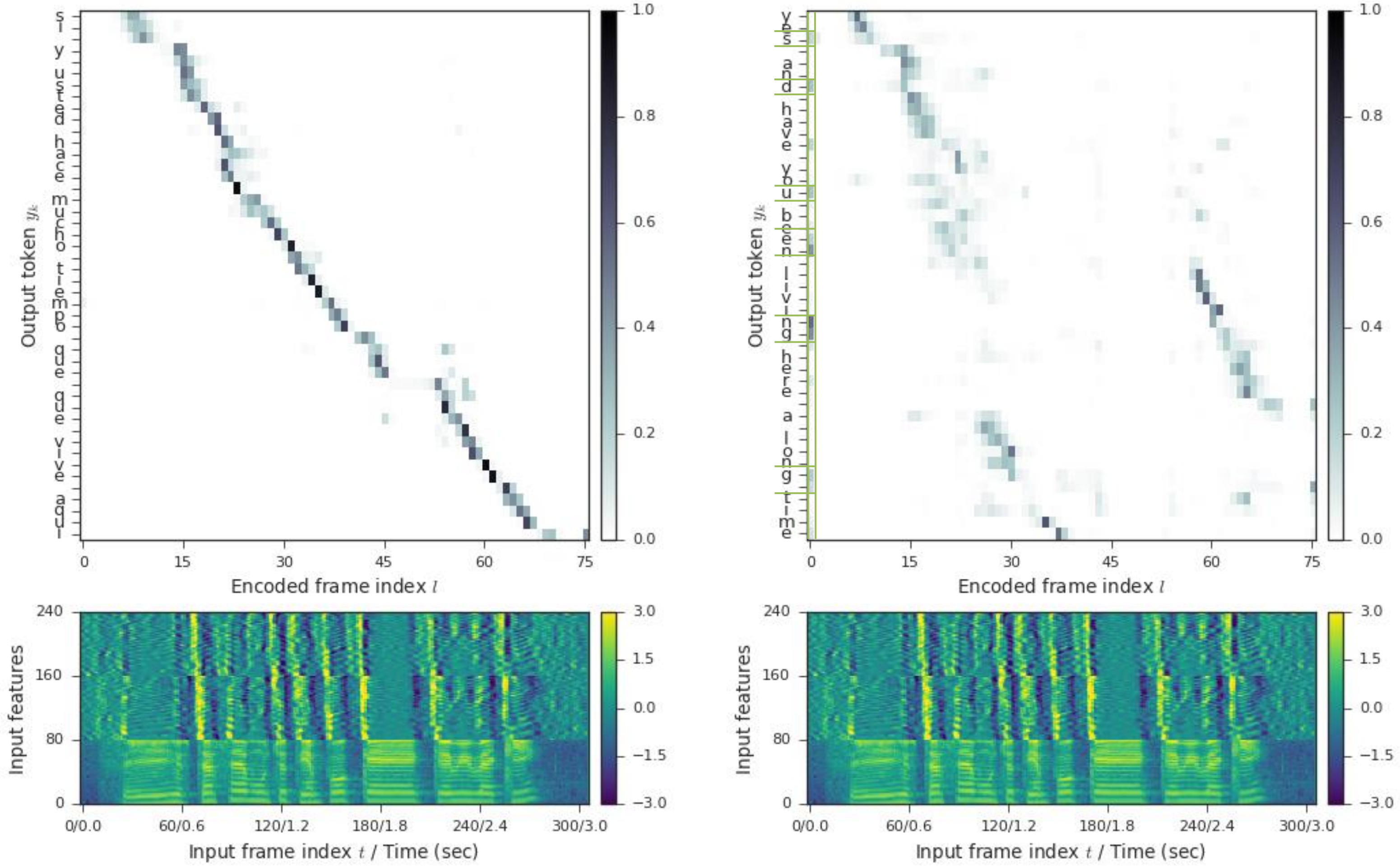
Seq2seq Speech Translation: Attention



- 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



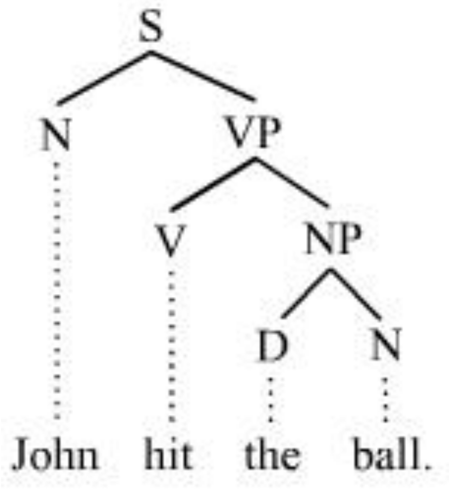
translation model attends to the beginning of input (i.e. silence) for the last few letters in each word

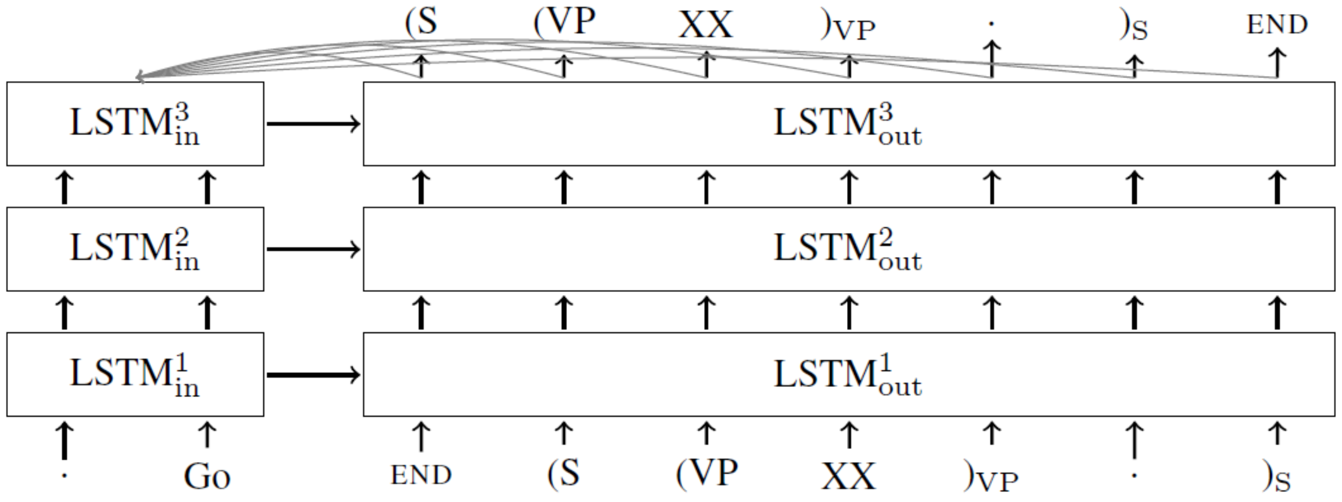
- already made a decision about word to emit, just acts a language model to spell it out.

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

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

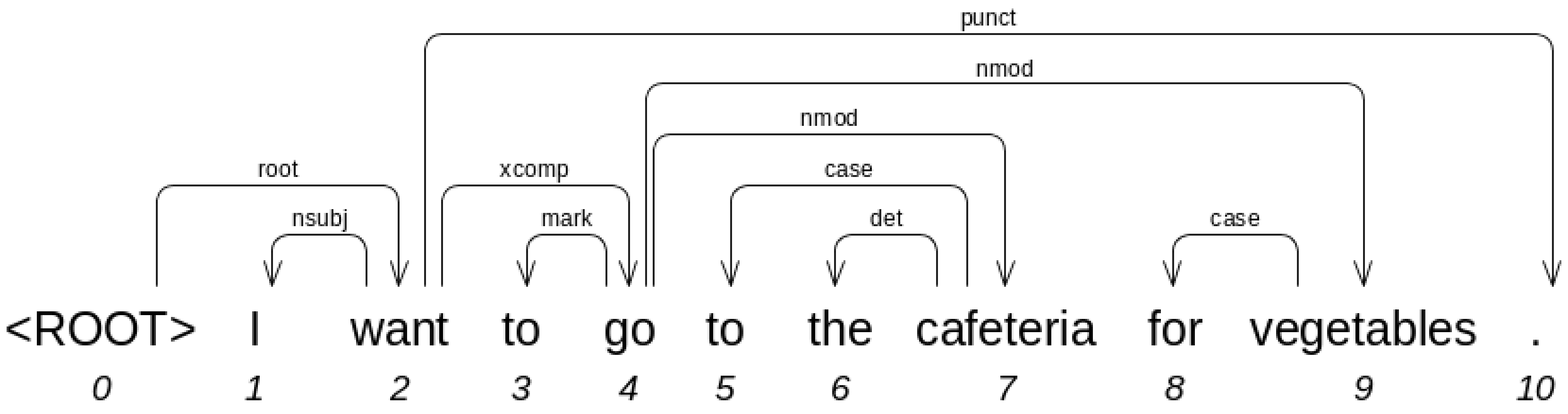
- Input: sentence
  „Go ."
- Output: linearized parse tree:
  „(S (VP XX)VP .)S END"





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

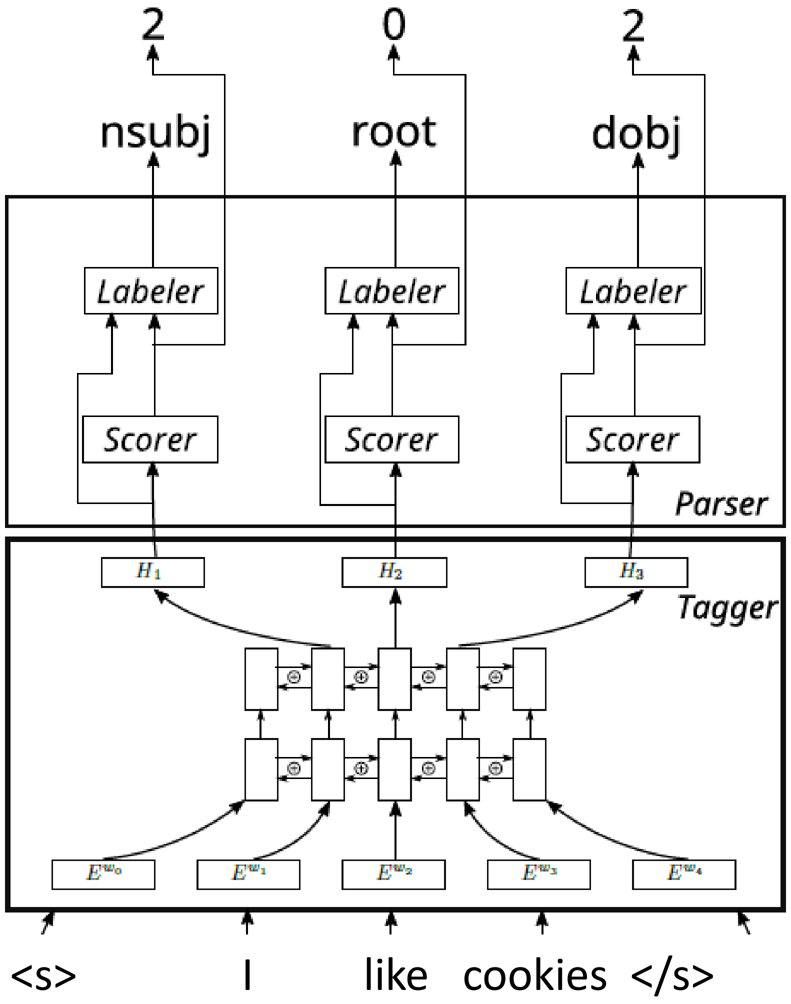
Dependency parsing



- 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

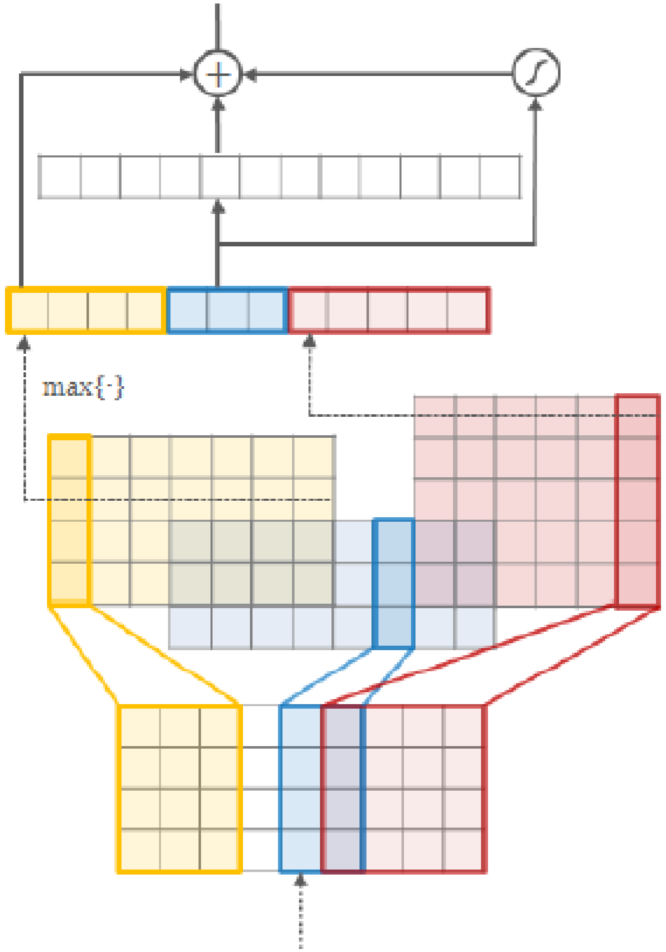


For each word w

Two operations:

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

From characters to word embeddings



„incomprehensible”

Highway layers – very nonlinear transformation of data

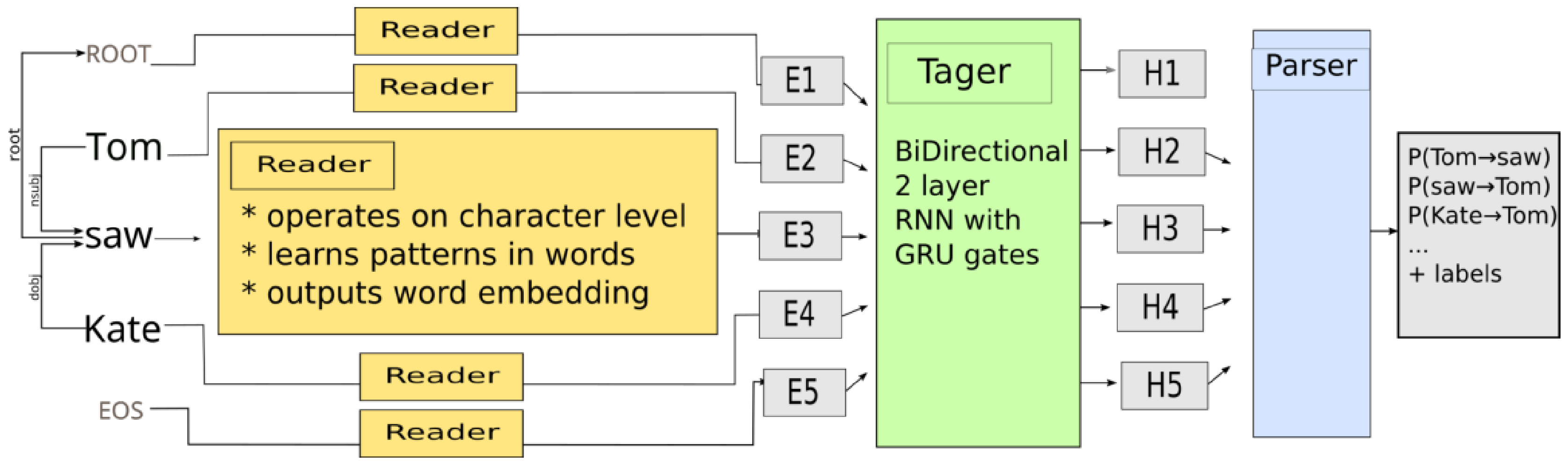
Glue best word representations together

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

Character embeddings concatenated into a matrix

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 непосредственным *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