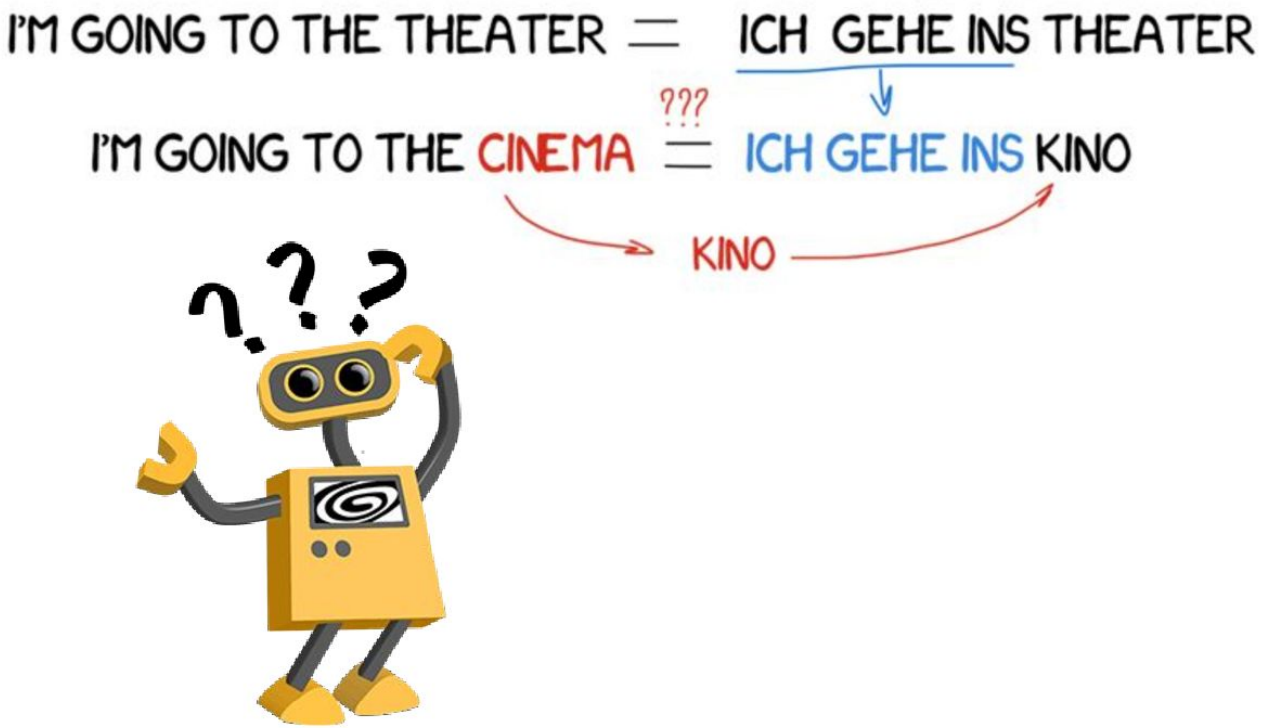
NLP 102

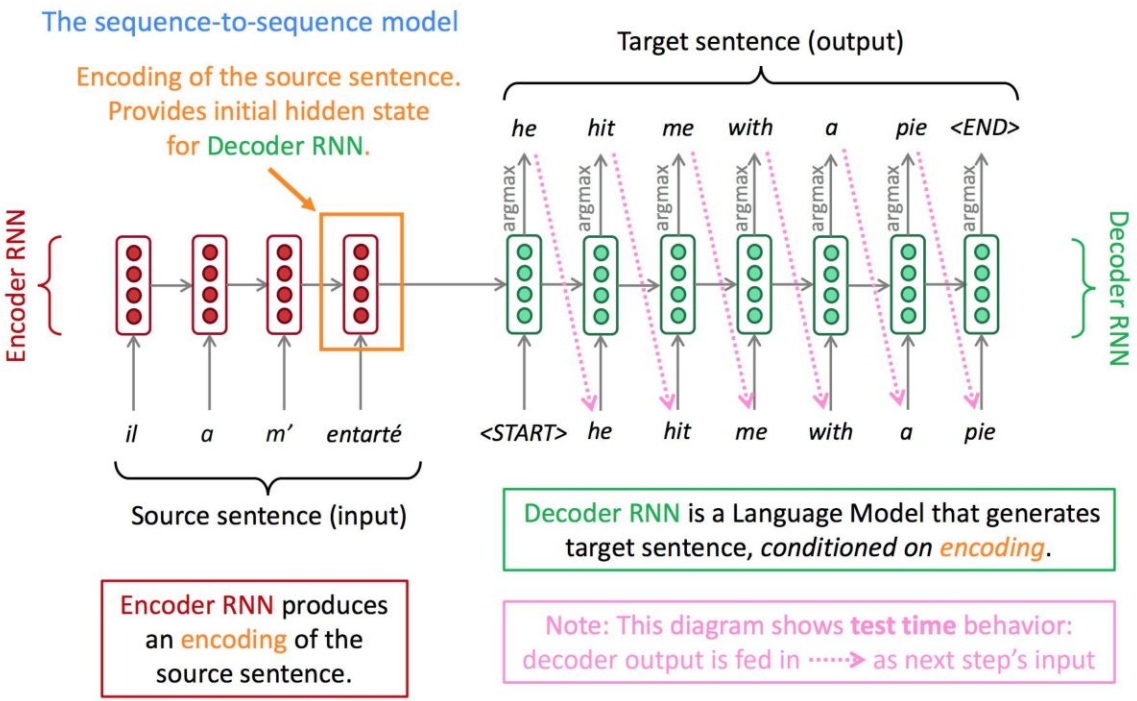
Attention on the slide

Тарас Хахулин
Skoltech, MIPT
Deep Learning Engineer, Samsung AI Center
Tg: @vitaminotar
https://github.com/khakhulin/

Machine Translation



Neural Machine Translation



NMT: how does it work?

- NMT directly calculates P(y|x) (y - target sentence, x - source sentence)

$$P(y|x) = P(y_1|x)\, P(y_2|y_1, x)\, P(y_3|y_1, y_2, x) \ldots \underbrace{P(y_T|y_1, \ldots, y_{T-1}, x)}$$

Probability of next target word, given target words so far and source sentence *x*

- To train it we need a huge parallel corpus.

Quality evaluation: BLEU

BLEU (Bilingual Evaluation Understudy) compares the machine-written translation to human-written translation, and computes a similarity score based on:

- n-gram precision
- penalty for too-short system translations

SYSTEM A: Israeli officials (2-GRAM MATCH) responsibility of airport (1-GRAM MATCH) safety

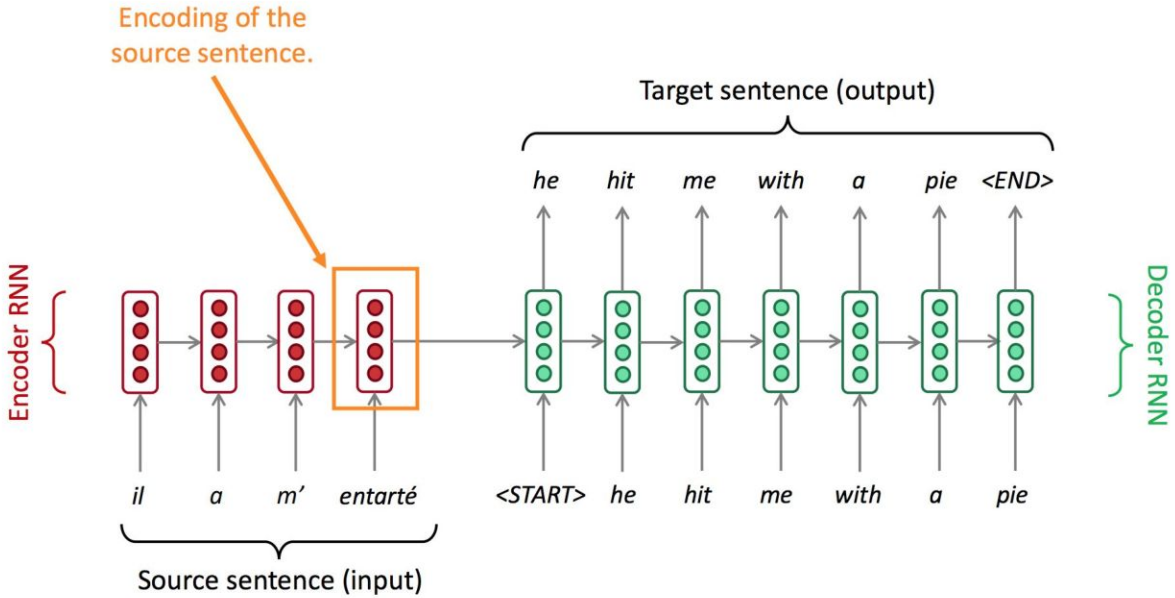
REFERENCE: Israeli officials are responsible for airport security

SYSTEM B: airport security (2-GRAM MATCH) Israeli officials are responsible (4-GRAM MATCH)

Metric	System A	System B
precision (1gram)	3/6	6/6
precision (2gram)	1/5	4/5
precision (3gram)	0/4	2/4
precision (4gram)	0/3	1/3
brevity penalty	6/7	6/7
BLEU	0%	52%

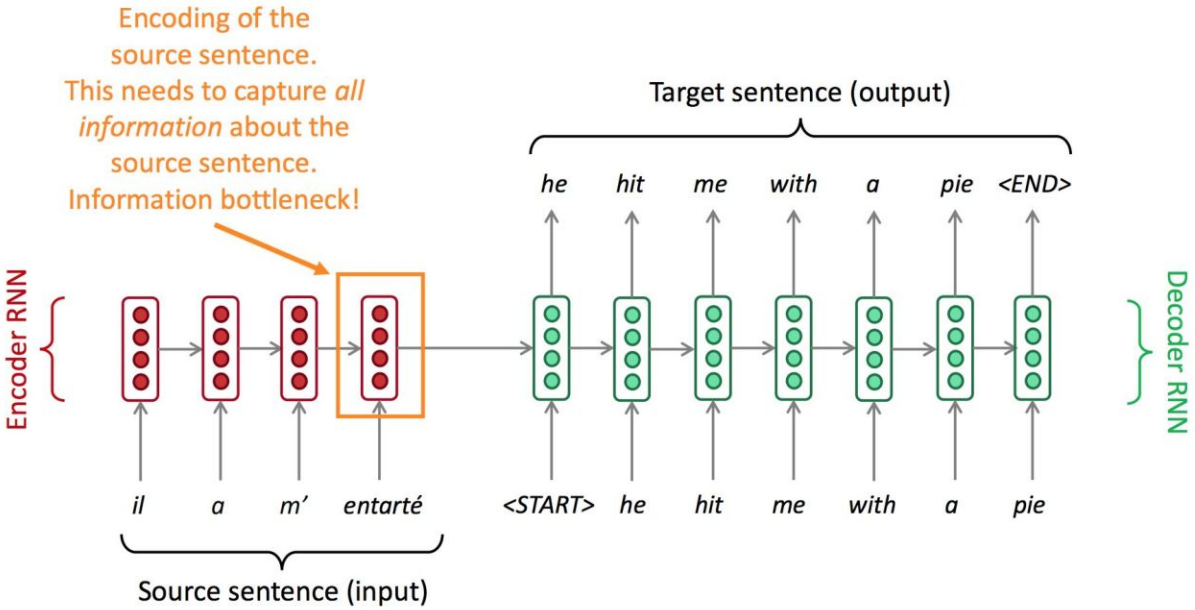
Attention

Seq2seq: the bottleneck problem



Problems with this architecture?

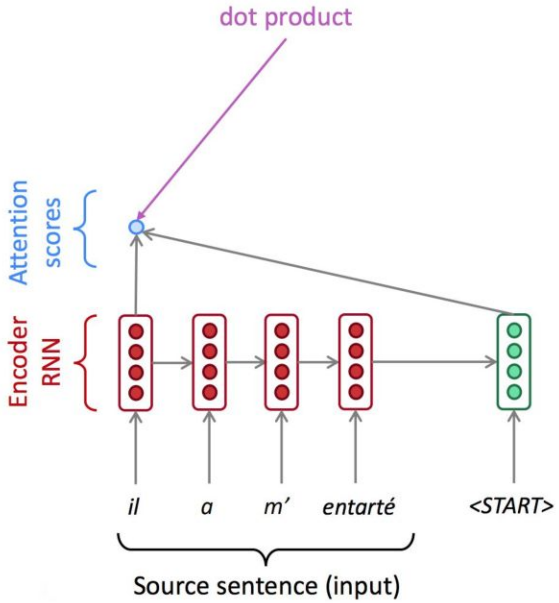
Seq2seq: the bottleneck problem



Attention

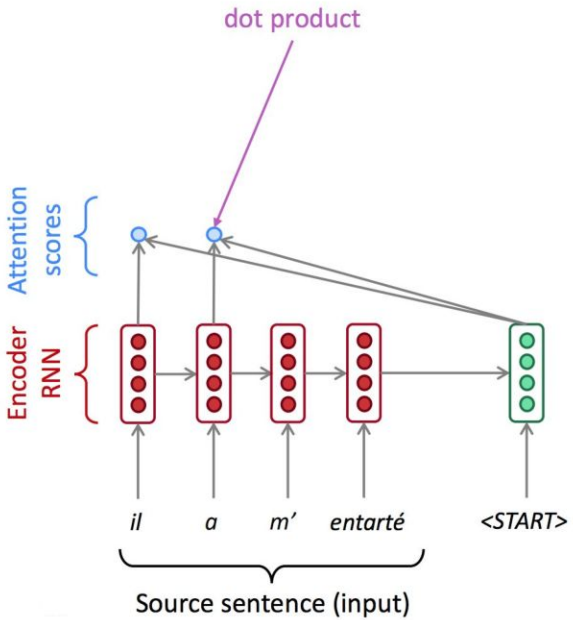
Main idea: on each step of the **decoder**, use **direct connection to the encoder** to focus on a particular part of the source sequence

Seq2seq with attention



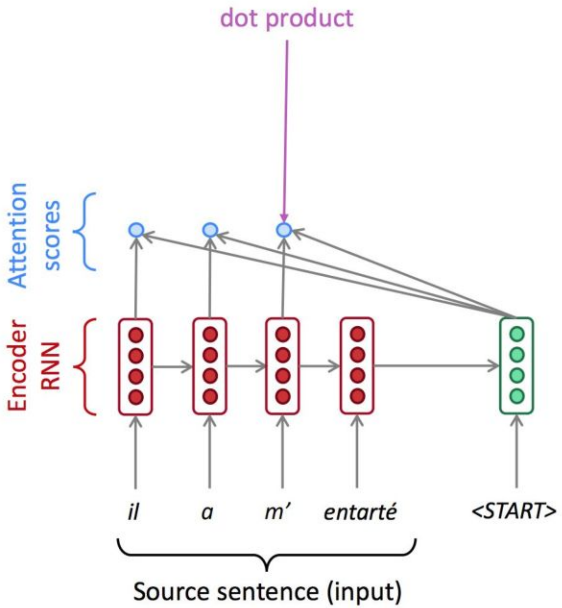
Decoder RNN

Seq2seq with attention

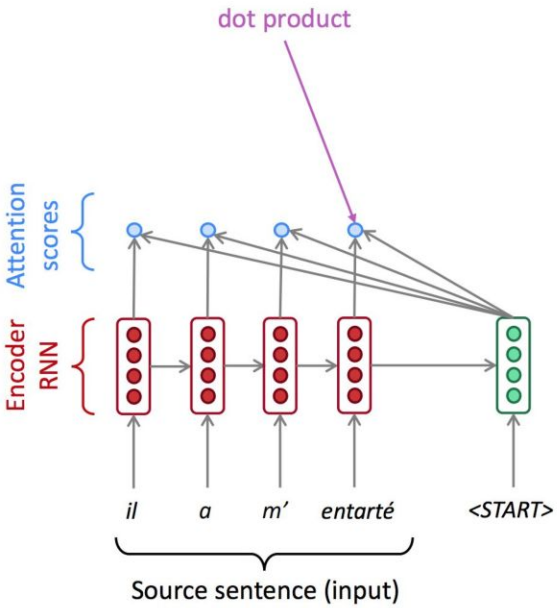


Decoder RNN

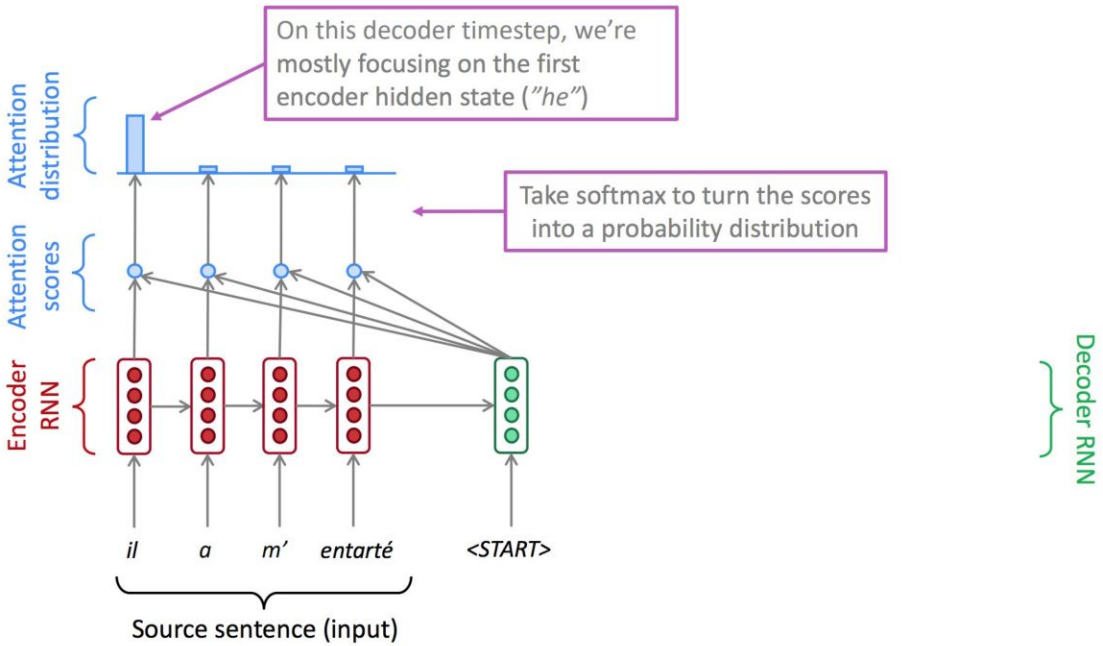
Seq2seq with attention



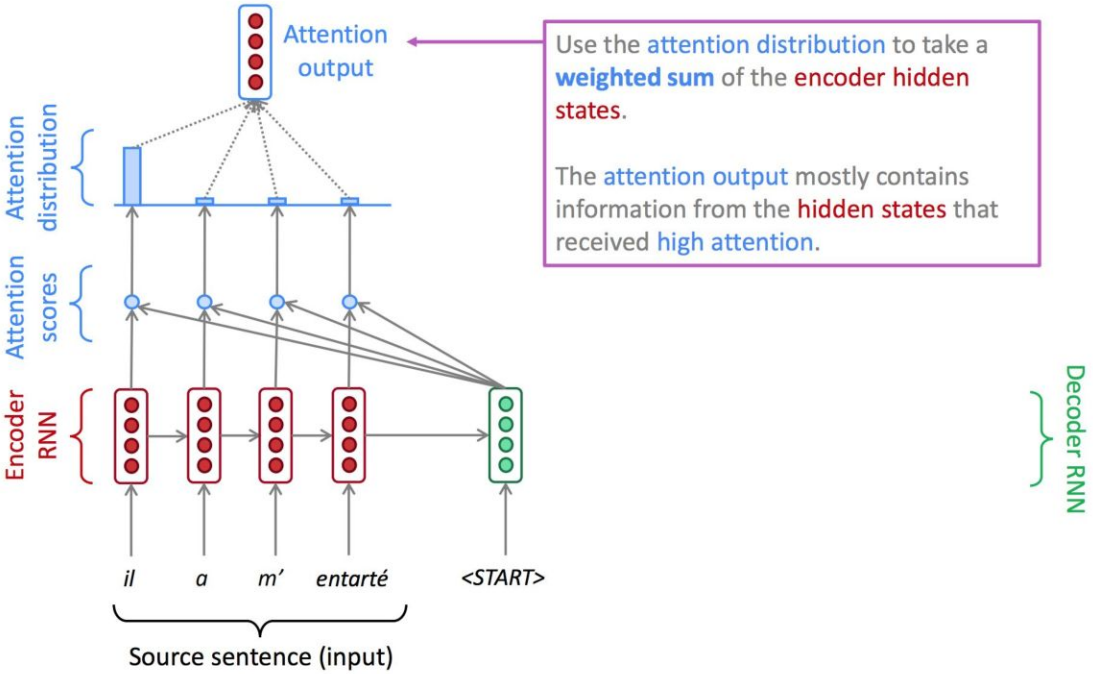
Seq2seq with attention



Seq2seq with attention



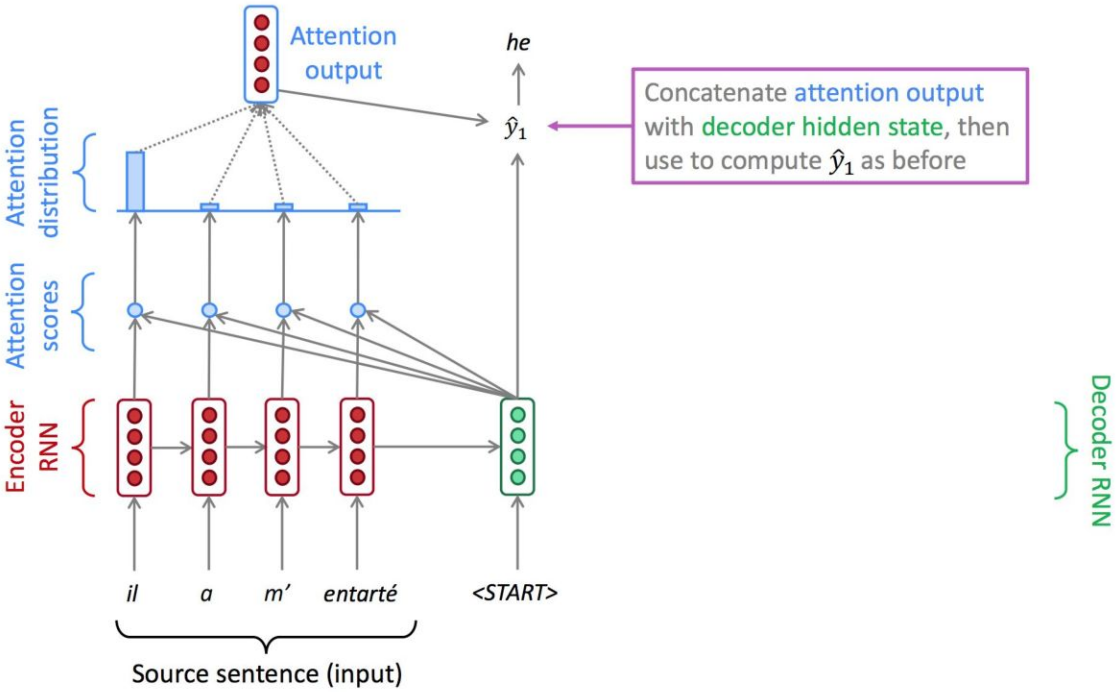
Seq2seq with attention



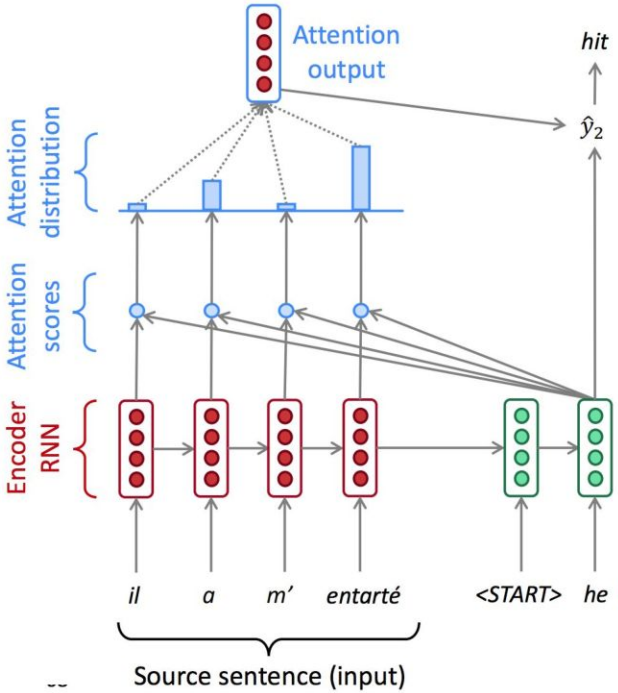
Use the attention distribution to take a **weighted sum** of the encoder hidden states.

The attention output mostly contains information from the hidden states that received high attention.

Seq2seq with attention



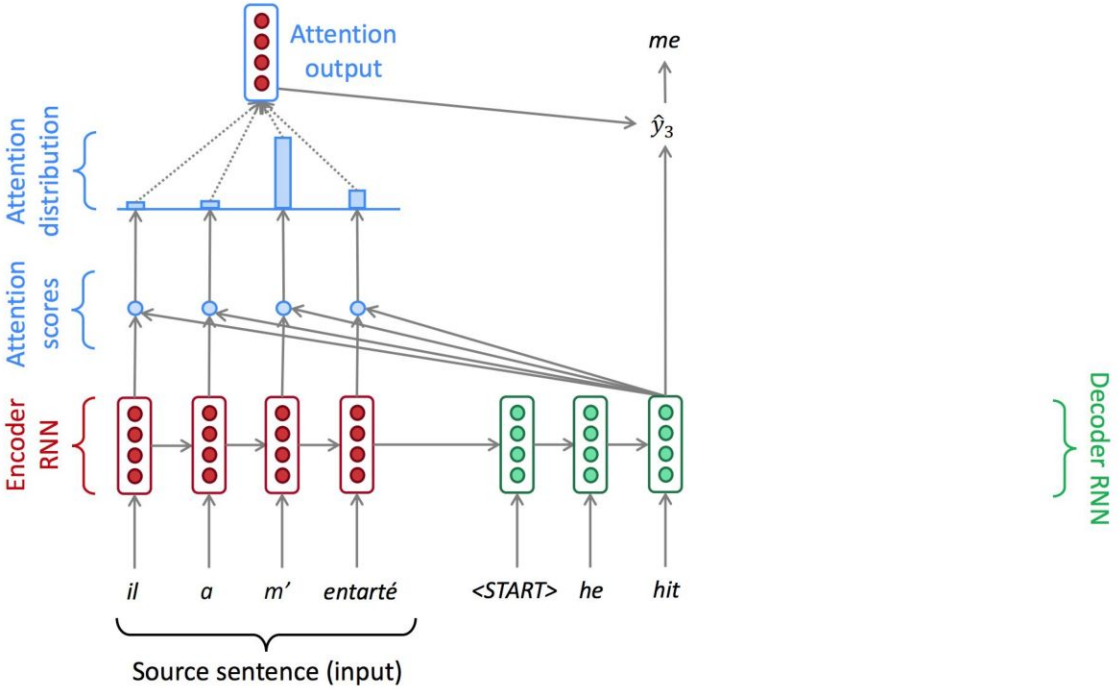
Seq2seq with attention



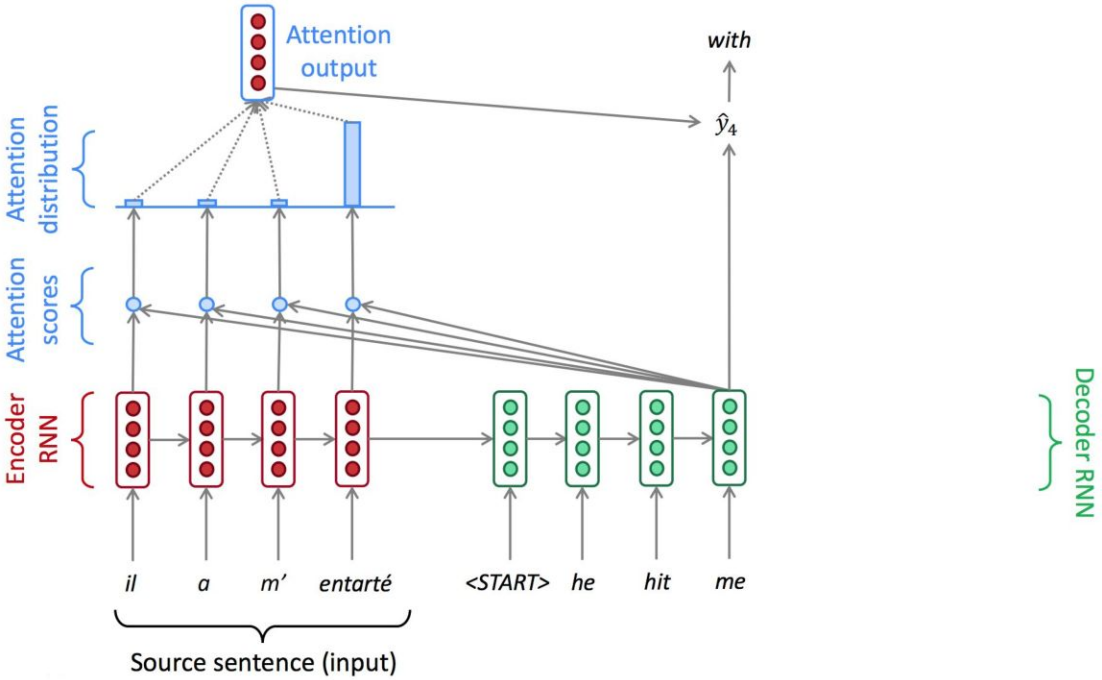
Sometimes we take the attention output from the previous step, and also feed it into the decoder (along with the usual decoder input). We do this in Assignment 4.

Decoder RNN

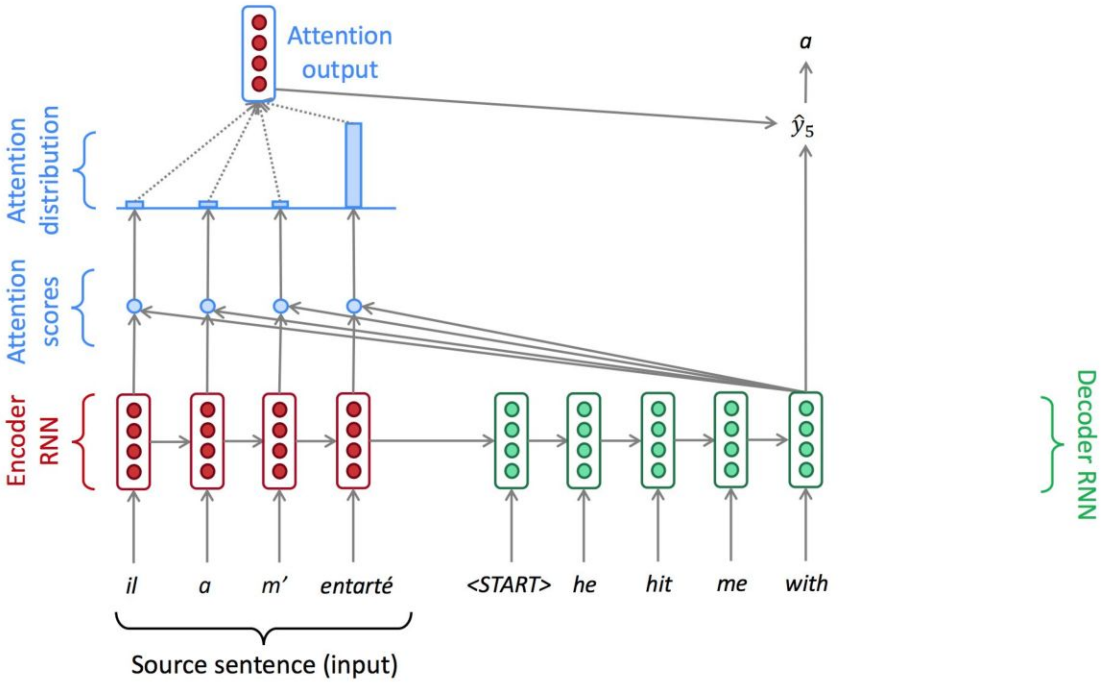
Seq2seq with attention



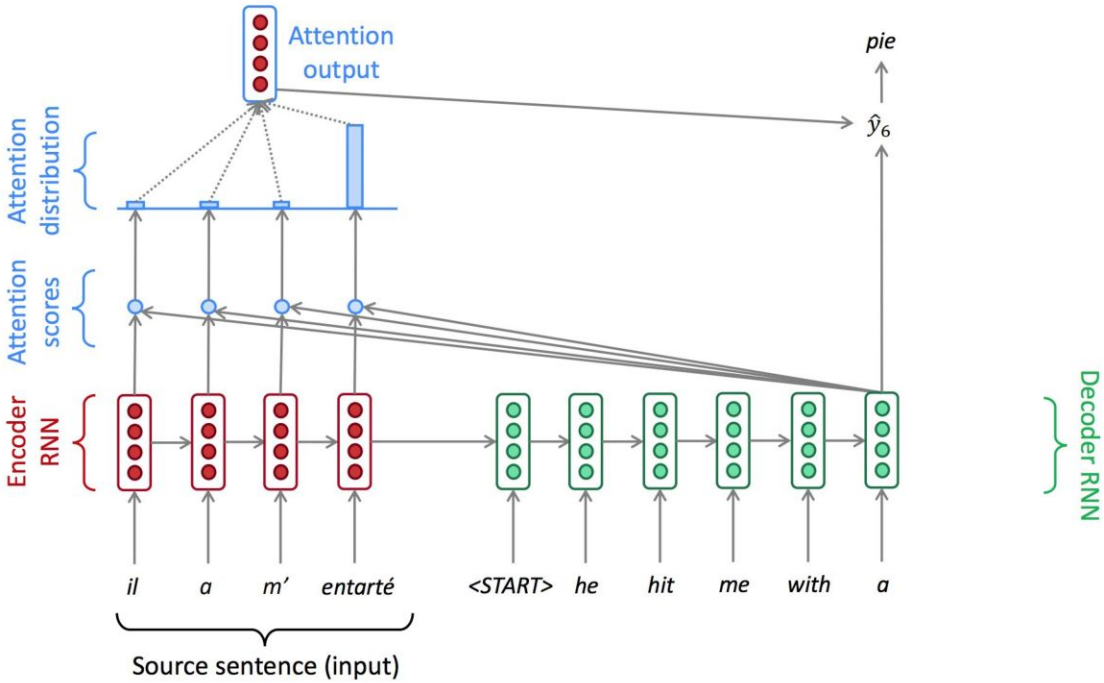
Seq2seq with attention



Seq2seq with attention



Seq2seq with attention



Attention in equations

dot product

bilinear function

multi-layer perceptron

any, literally, ANY function you can imagine

$$f_{att}(h_i, s_j) = h_i^\top s_j$$

$$f_{att}(h_i, s_j) = h_i^\top \mathbf{W}_a s_j$$

$$f_{att}(\mathbf{h}_i, \mathbf{s}_j) = \mathbf{v}_a{}^\top \tanh(\mathbf{W}_a[\mathbf{h}_i; \mathbf{s}_j])$$

$$z(c, m, q) = \left[c, m, q, c \circ q, c \circ m, |c - q|, |c - m|, c^T W^{(b)} q, c^T W^{(b)} m\right]$$

$$G(c, m, q) = \sigma\left(W^{(2)} \tanh\left(W^{(1)} z(c, m, q) + b^{(1)}\right) + b^{(2)}\right)$$

A bit crazy, huh?

Attention in equations

Self-Attention(&)	Relating different positions of the same input sequence. Theoretically the self-attention can adopt any score functions above, but just replace the target sequence with the same input sequence.
Global/Soft	Attending to the entire input state space.
Local/Hard	Attending to the part of input state space; i.e. a patch of the input image.

What is Hard?

Soft Attention: the alignment weights are learned and placed “softly” over all patches in the source image;

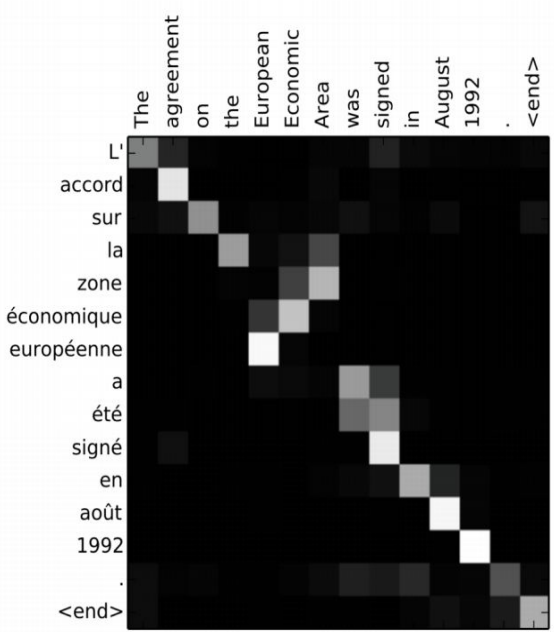
- (+) the model is smooth and differentiable
- (-) expensive when the source input is large

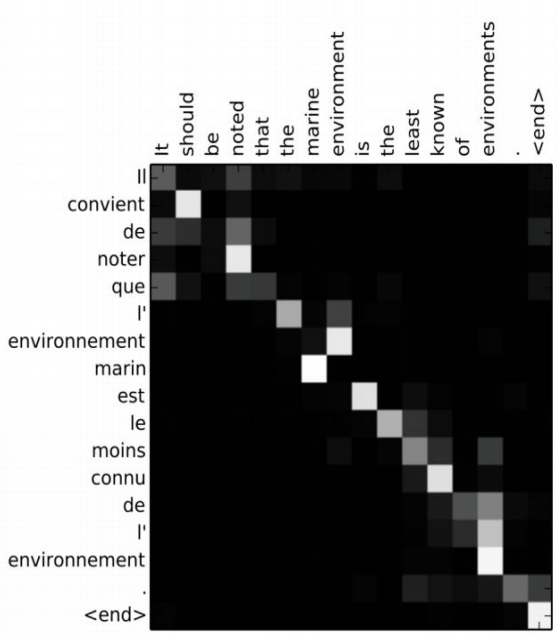
Hard Attention: only selects one patch of the image to attend to at a time

- (+) less calculation at the inference time
- (-) model is non-differentiable and requires more complicated techniques such as variance reduction or reinforcement learning to train

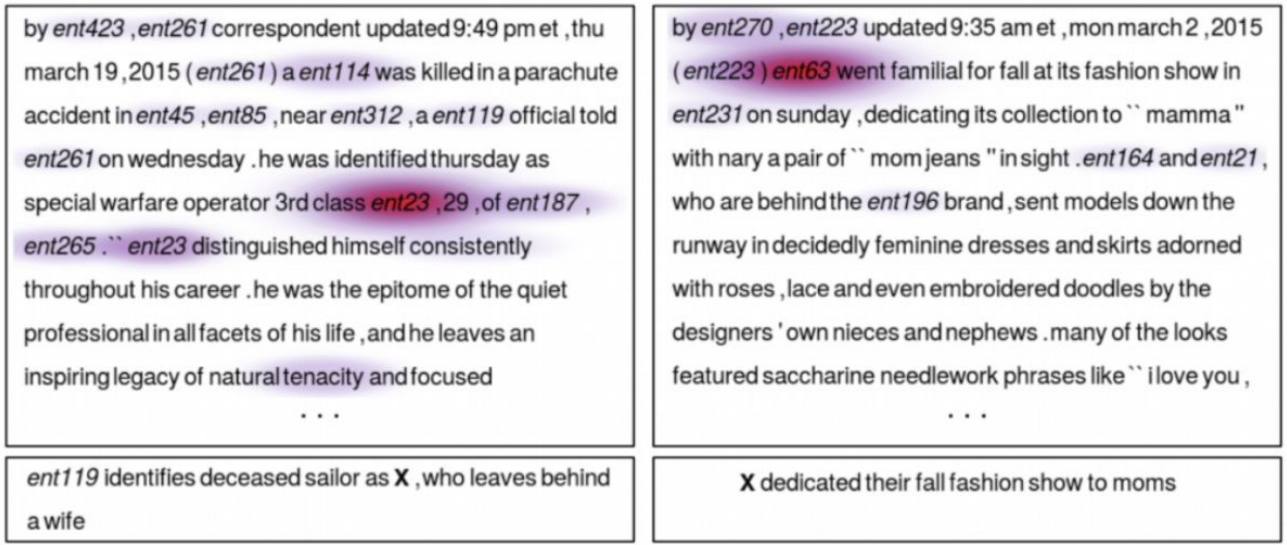
Attention provides interpretability

- We may see what the decoder was focusing on
- We get word alignment for free!





Question Answering task



Use a RNN to read a text, read a (synthetically generated) question, and then produce an answer.

Attention variants

- Basic dot-product (the one discussed before): $e_i = s^T h_i \in \mathbb{R}$
- Multiplicative attention: $e_i = s^T W h_i \in \mathbb{R}$
  - $W \in \mathbb{R}^{d_2 \times d_1}$ - weight matrix
- Additive attention: $e_i = v^T \tanh(W_1 h_i + W_2 s) \in \mathbb{R}$
  - $W_1 \in \mathbb{R}^{d_3 \times d_1}, W_2 \in \mathbb{R}^{d_3 \times d_2}$ - weight matrices
  - $v \in \mathbb{R}^{d_3}$ - weight vector

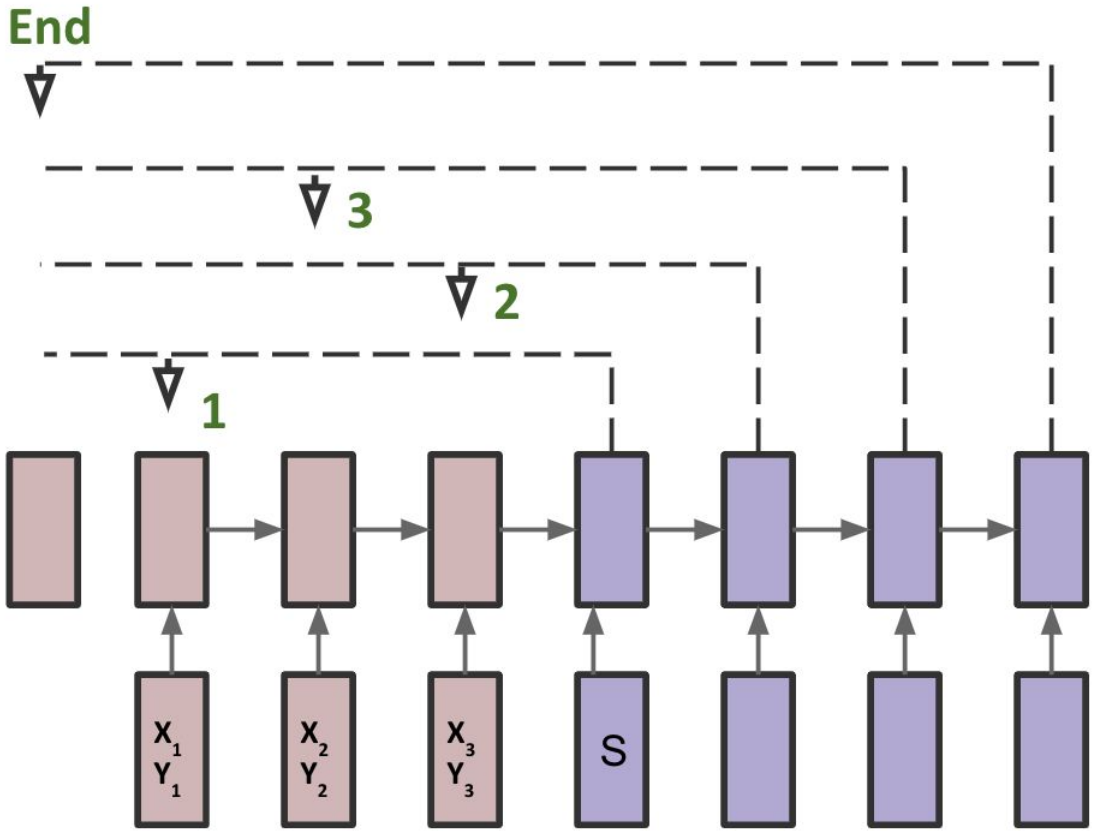
Pointer Networks

$$p\left(\pi_i|\pi_{1:i-1}, \pi_{1:T}\right) = \text{softmax}\left(s_{i,j}\right)$$

where

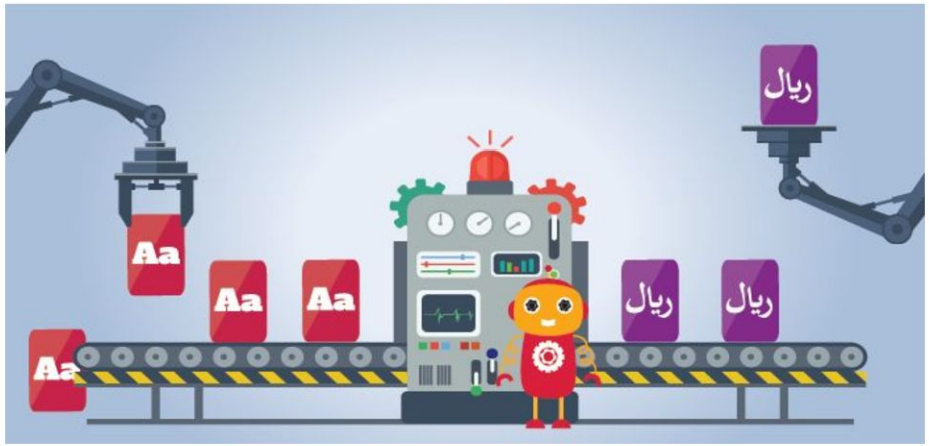
$$s_{i,j} = v^T \tanh(U_1 x_j + U_2 h_i), \forall j \in [1, T]$$

RL etc.



Summary

- Seq2seq is an architecture for NMT (2 RNNs)
- Attention is a way to focus on particular parts of the input



Bonus: Dialog system parts

Task-oriented dialog system

You can talk to a personal assistant:

- Apple Siri
- Google Assistant
- Microsoft Cortana
- Amazon Alexa
- …

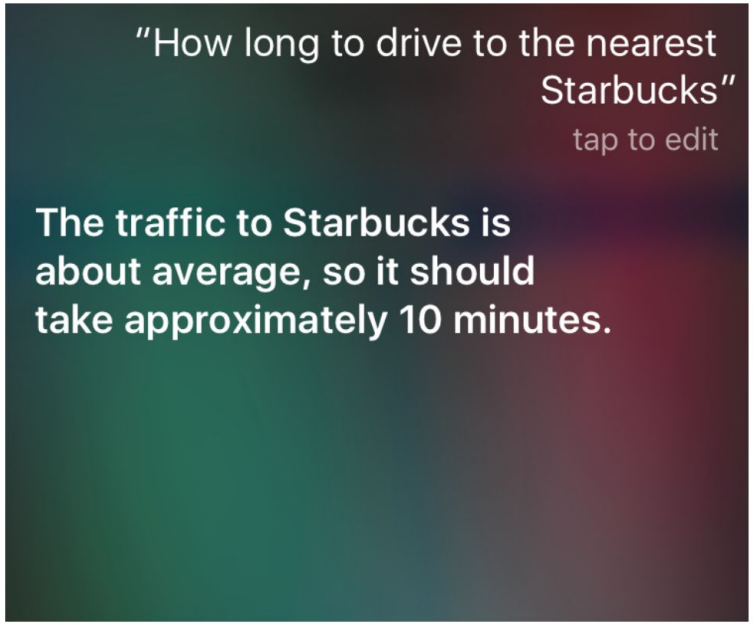
You can solve these tasks:

- Set up a reminder
- Find photos of your pet
- Find a good restaurant
- Send a message
- …

https://www.coursera.org/learn/language-processing

Intent classification

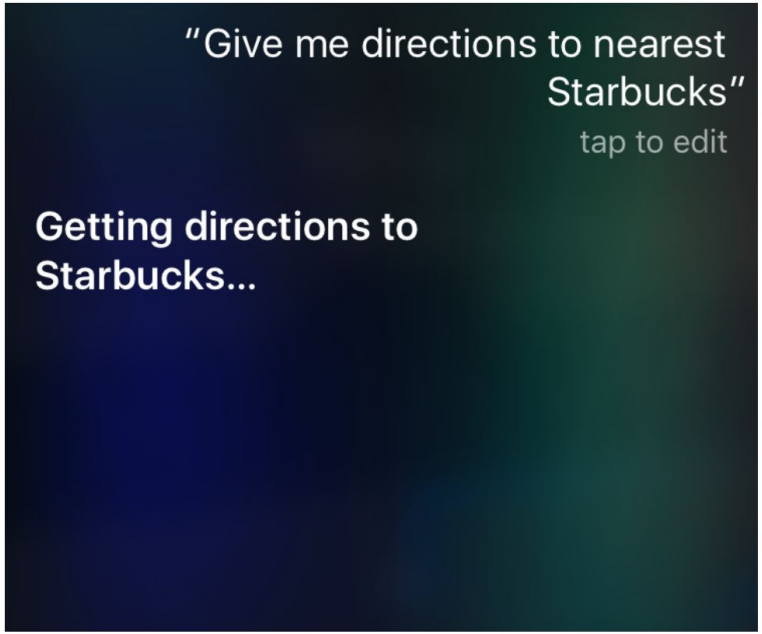
- What does the user want?
- Which predefined scenario is the user trying to execute?



Intent: **nav.time.closest**

There’re many intents

- And you need to classify them to give correct answers
- This is a classification task and you can measure **accuracy**



Intent: **nav.directions.closest**

Form filling approach to dialog management

- Think of an intent as a **form** that a user needs to fill in.
- Each intent has a set of fields (**slots**) that must be filled in to execute the request.

- Example: **nav.directions** intent
  - **@FROM** slot: defaults to current geolocation
  - **@TO** slot: required

- We need a **slot tagger** to extract slots from utterance.

Slot filling/tagging

- Example:
  - User: Show me the way to History Museum
  - Slot tagger: Show me the way to @TO{History Museum}
- This is token classification task in **BIO** scheme:
  - **B** corresponds to the word at the **beginning** of the slot
  - **I** corresponds to the word **inside** the slot (excluding beginning)
  - **O** corresponds to words **outside** of slots
- Example of **BIO** scheme tagging:

Tokens	Show	me	the	way	to	History	Museum
Tags	O	O	O	O	O	B-TO	I-TO

Form filling dialog manager (single turn)

- User: **Give me directions to San Francisco**
  - Intent classifier: nav.directions
  - Slot tagger: @TO{San Francisco}
  - Dialog manager: *all slots are filled, here's the route*
- Agent (assistant): **Here's the route**

Google Maps

Form filling dialog manager (multi-turn)

- User: **Give me directions from Los Angeles**
  - Intent classifier: nav.directions
  - Slot tagger: @FROM{Los Angeles}
  - Dialog manager: *required slot is missing, where to?*
- Agent (assistant): **Where do you want to go?**
- User: **San Francisco**
  - Intent classifier: nav.directions
  - Slot tagger: @TO{San Francisco}
  - Dialog manager: *okay, here's the route*
- Agent (assistant): **Here's the route**

Google Maps

Form filling dialog manager (multi-turn)

- User: **Give me directions from Los Angeles**
  - Intent classifier: nav.directions
  - Slot tagger: @FROM{Los Angeles}
  - Dialog manager: *required slot is missing, where to?*
- Agent (assistant): **Where do you want to go?**
- User: **San Francisco**
  - Intent classifier: nav.directions
  - Slot tagger: @TO{San Francisco}
  - Dialog manager: *okay, here's the route*
- Agent (assistant): **Here's the route**

We need context here

Google Maps

How to track context (an easy way)

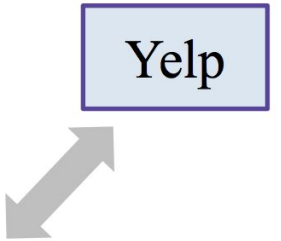
- Both intent classifier and slot tagger need context (what happened before)

- Let's add simple **features** to both of them:
  - Previous utterance intent as a categorical feature
  - Slots filled in so far with binary feature for each possible slot
- Improves slot tagger F1 by 0.5%
- Reduces intent classifier error by 6.7%

$F = 2PR/(P+R)$

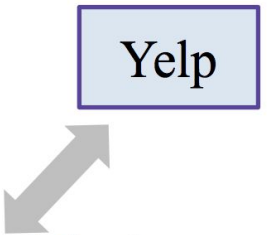
How to track a form switch

- User: **Give me directions from Los Angeles**
  - Intent classifier: nav.directions
  - Slot tagger: @FROM{Los Angeles}
  - Dialog manager: *required slot is missing, where to?*
- Agent (assistant): **Where do you want to go?**
- User: **Forget about it, let's eat some sushi first**
  - Intent classifier: nav.find
  - Slot tagger: @CATEGORY{sushi}
  - Dialog manager: *okay, let's start a new form and find some sushi*
- Agent (assistant): **Okay, here are nearby sushi places**

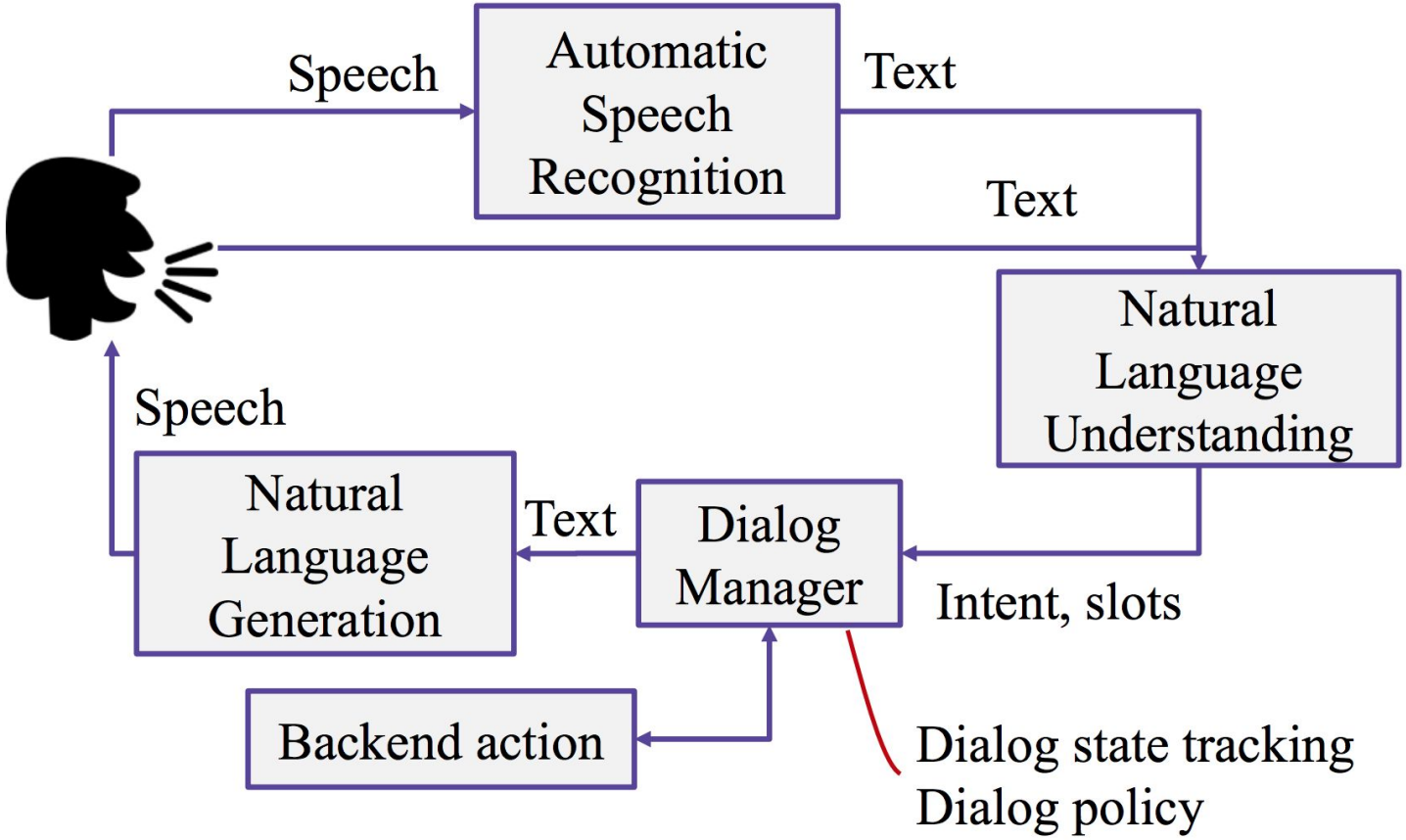


How to track a form switch

- User: **Give me directions from Los Angeles**
  - Intent classifier: nav.directions
  - Slot tagger: @FROM{Los Angeles}
  - Dialog manager: *required slot is missing, where to?*
- Agent (assistant): **Where do you want to go?**
- User: **Forget about it, let's eat some sushi first**
  - Intent classifier: nav.find
  - Slot tagger: @CATEGORY{sushi}
  - Dialog manager: *okay, let's start a new form and find some sushi*
- Agent (assistant): **Okay, here are nearby sushi places**



Task-oriented dialog system overview



Intent classifier

- What you can do:
  - Any model on BOW with n-grams and TF-IDF
  - RNN (LSTM, GRU, …)
  - CNN (1D convolutions)

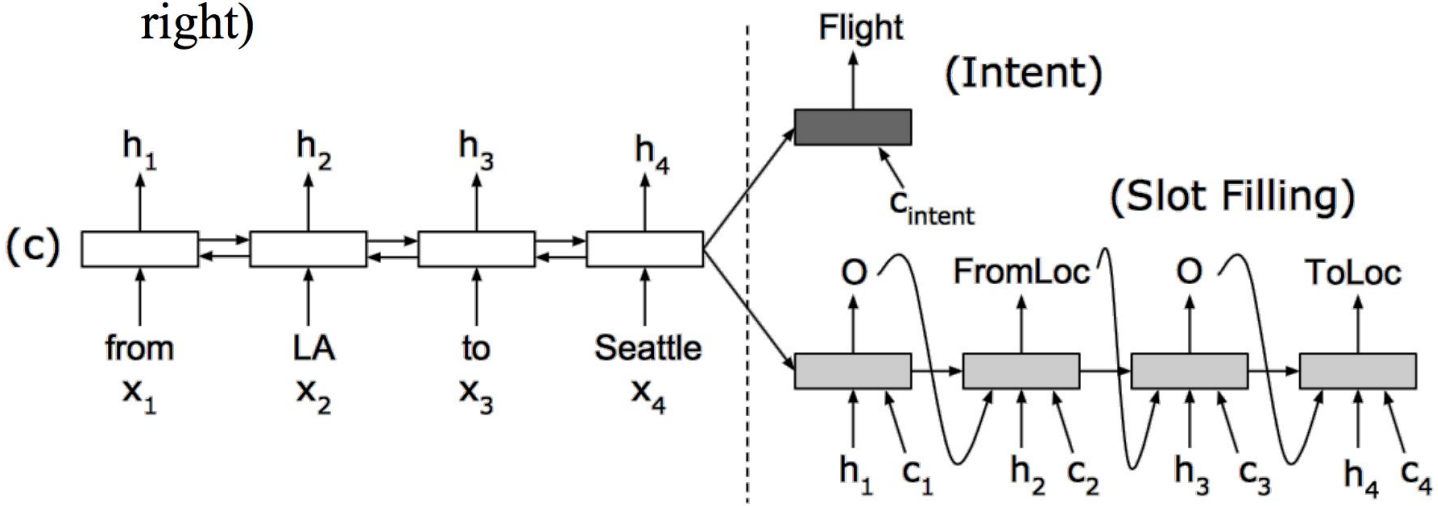
- CNNs can perform better on datasets where the task is essentially a key phrase recognition task as in some sentiment detection datasets.

Slot tagger

- What you can do:
  - Handcrafted rules like regular expressions
  - CRF
  - RNN seq2seq
  - **CNN seq2seq**
  - Any seq2seq with attention

Joint training of intent classifier and slot tagger

- Encoder-decoder architecture for joint intent detection and slot filling
- Encoder is a bi-directional LSTM
- With aligned inputs (h_i on the right) and attention (c_i on the right)



Links

More on RNN

- Distill.pub post on attention and augmentations for RNN - post
- Seq2seq lecture - video
- BLEU and CIDEr articles.
- Image captioning
  - MSCOCO captioning challenge
  - Captioning baseline notebook
- Lecture on attention mechanisms - video (RUSSIAN)
- Stanford lecture on seq2seq and MT (english) - video

More Language Models

- CS231 lecture on RNNs - video (english)
- A more detailed lecture by Y. Bengio - video
- Great reading by A. Karpathy - http://karpathy.github.io/2015/05/21/rnn-effectiveness/
- LSTM explained in detail by colah - http://colah.github.io/posts/2015-08-Understanding-LSTMs/
- "Awesome rnn" entry point - https://github.com/kjw0612/awesome-rnn

Two original NMT papers:

[1] Sutskever et al, NIPS 2014, https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

[2] Bahdanau et al, ICLR 2015, https://arxiv.org/pdf/1409.0473.pdf

Links

Multilingual Embeddings. Unsupervised MT.

- Exploiting similarities between languages for machine translation Mikolov et al., 2013 [arxiv]
- Improving vector space word representations using multilingual correlation Faruqui and Dyer, EACL 2014 [pdf]
- Learning principled bilingual mappings of word embeddings while preserving monolingual invariance Artetxe et al., EMNLP 2016 [pdf]
- Offline bilingual word vectors, orthogonal transformations and the inverted softmax [arxiv] Smith et al., ICLR 2017
- Word Translation Without Parallel Data Conneau et al., 2018 [arxiv]

More materials about Embeddings

- On hierarchical & sampled softmax estimation for word2vec page
- GloVe project page
- FastText project repo
- Semantic change over time - oberved through word embeddings - arxiv