Sequence-to-sequence models

used for machine translation and

Murat Apishev
Katya Artemova

Computational Pragmatics Lab, HSE

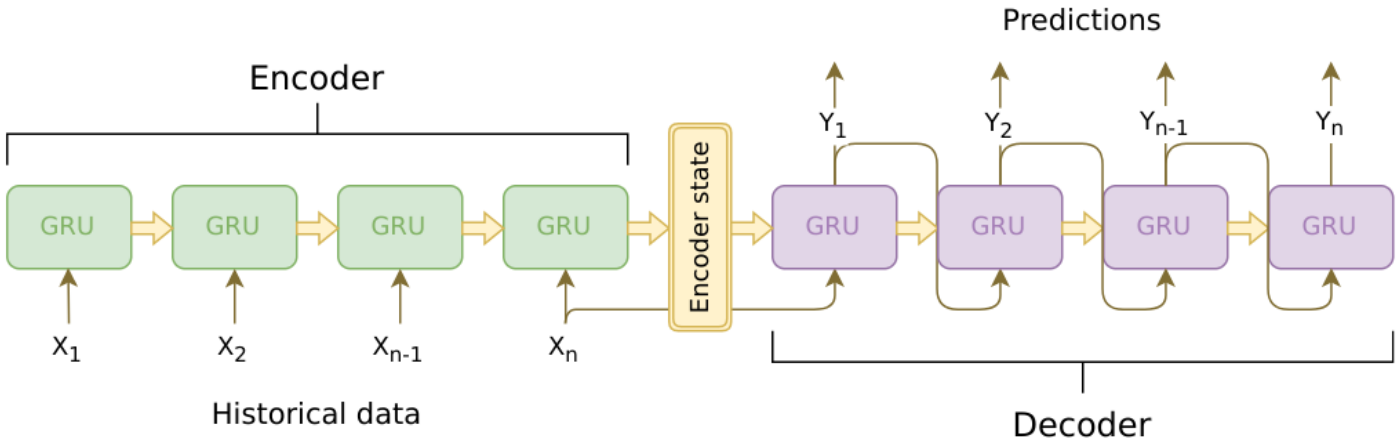
December 2, 2019

Today

Sequence-to-sequence

Neural encoder encoder architectures

- Achieves high results on machine translation, spelling correction, summarization and other NLP tasks
- The encoder inputs sequence of tokens $x_{1:n}$ and outputs hidden states h_n^E
- The decoder decodes an output sequence of tokens $y_{1:n}$by decoding last hidden state $h_0^D = h_n^E$
- seq2seq architectures are trained on **parallel corpora**



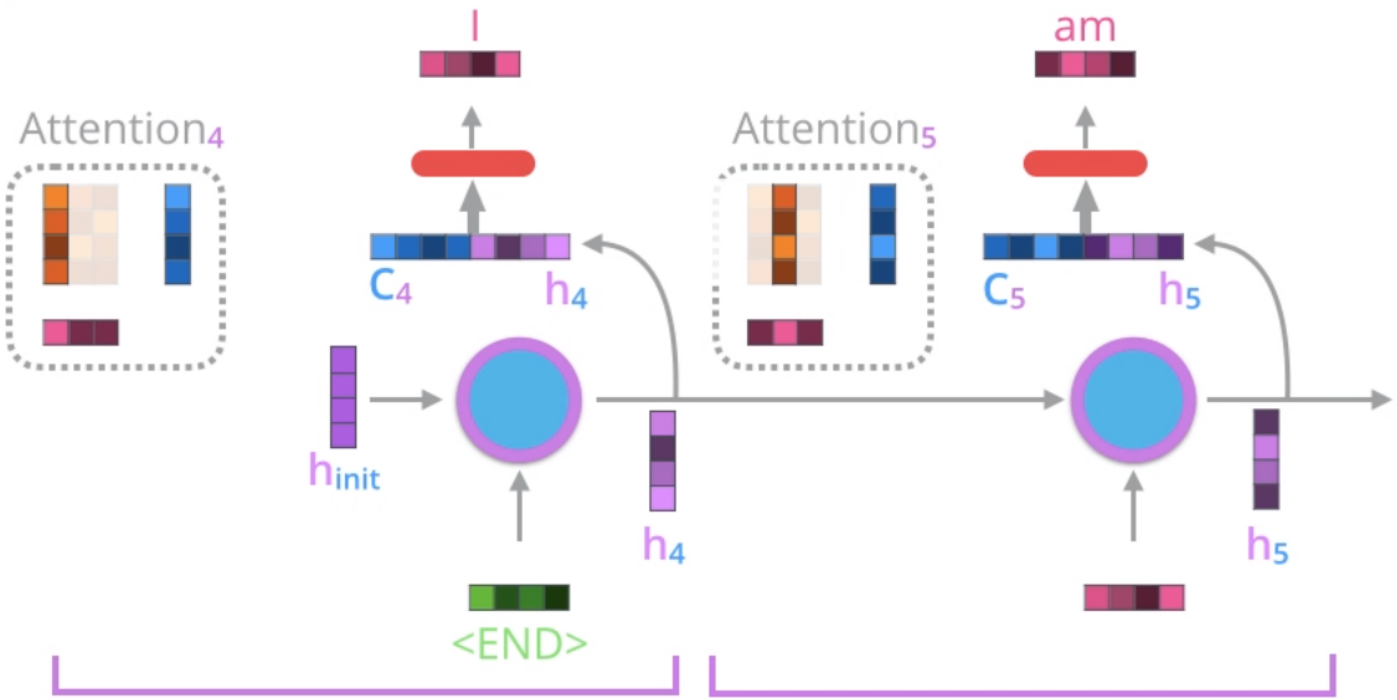
Seq-to-seq for MT

- Both encoder and decoder are recurrent networks
- Input words x_i ($i \in [1, n]$) are represented as word embeddings (w2v for example)
- The context vector: h_n, last hidden state of RNN encoder, turns out to be a bottleneck
- It is challenging for the models to deal with long sentences as the impact of last words is higher
- **Attention mechanism** is one of the possible solutions

Seq-to-seq for MT + attention

Attention mechanism allows to align input and output words.

- The encoder passes all the hidden states to the decoder: not h_n^E, but rather $h_i^E, i \in [1, n]$
- The hidden states can be treated as context-aware word embeddings
- The hidden states are used to produced a context vector c for the decoder



Seq-to-seq for MT + attention

- At the step j the decoder inputs $h^D_{j-1}, j \in [n+1, m]$ and a context vector c_j from the encoder
- The context vector c_j is a linear combination of the encoder hidden states:

$$c_j = \sum_i \alpha_i h_i^E$$

- α_i are attention weights which help the decoder to focur on the relvant part of the encoder input

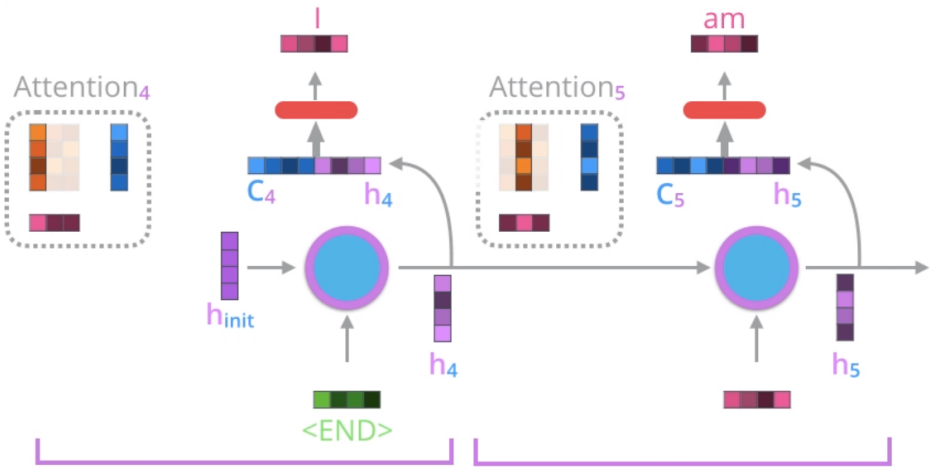
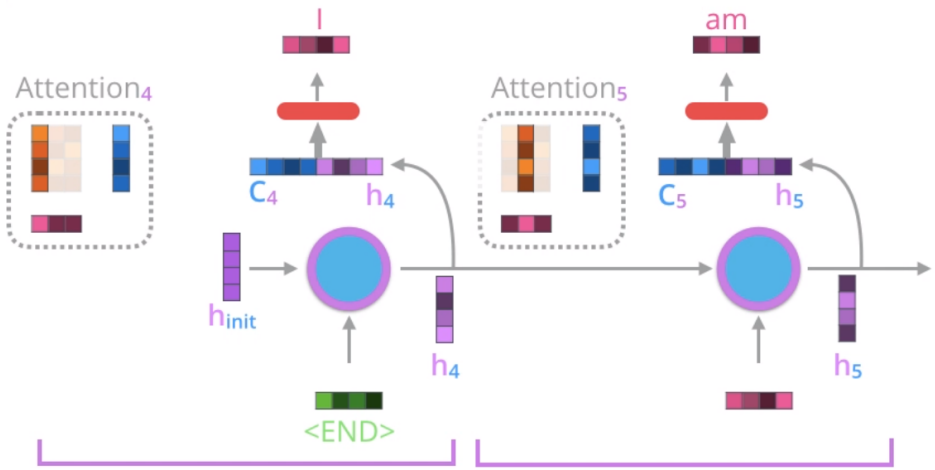


Image source: jalammar

Seq-to-seq MT + attention

To generate a new word the decoder at the step j:

- inputs h^D_{j-1} and produces h^D_j
- concatenates h^D_j to c_j
- passes the concatenated vector through linear layer with softmax activation to get a probability distribution over target vocabulary



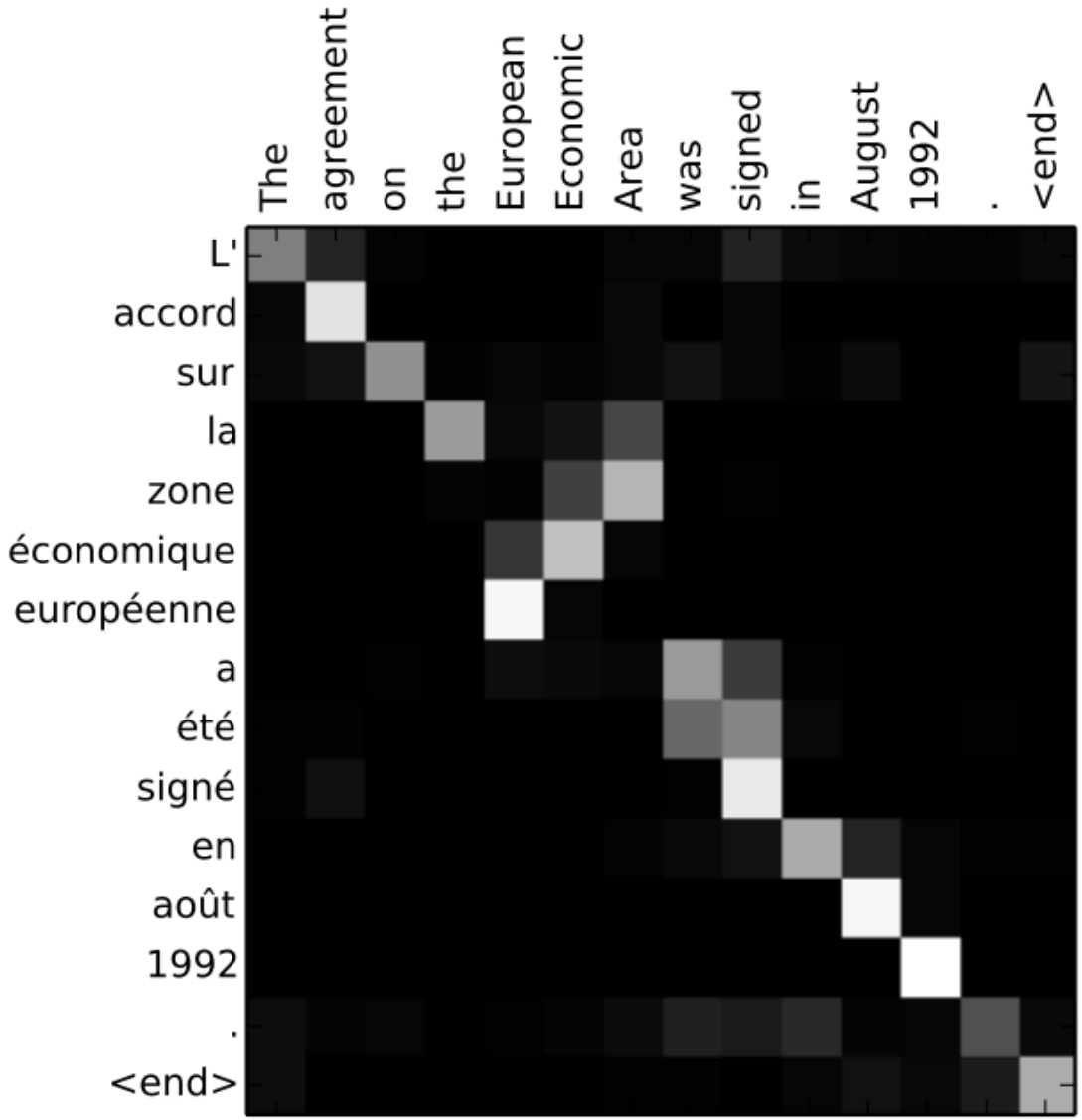
Attention weights

- The attention weights α_{ij} measure the similarity of the encoder hidden state h_i^E while generating the word j

$$a_{ij} = \frac{\exp(\mathrm{sim}(h_i^E, h_j^D))}{\sum_k \exp(\mathrm{sim}(h_k^E, h_j^D))}$$

- The similarity sim can be computed by
  - dot product attention:
  $$\mathrm{sim}(h, s) = h^T s$$
  - additive attention:
  $$\mathrm{sim}(h, s) = w^T \tanh(W_h h + W_s s)$$
  - multiplicative attention:
  $$\mathrm{sim}(h, s) = h^T W s$$
- Weights are trained jointly with the whole model

Attention map



MT metrics

- BLEU compares system output to reference translation

  Reference translation: *E-mail was sent on Tuesday*

  System output: *The letter was sent on Tuesday*

- Given N ($N \in [1, 4]$) compute the number of N-grams present both in system output and reference translation:

$$N = 1 \Rightarrow \frac{4}{6} \qquad N = 2 \Rightarrow \frac{3}{5} \qquad N = 3 \Rightarrow \frac{2}{4} \qquad N = 4 \Rightarrow \frac{1}{3}$$

- Take geometric mean N:

$$\text{score} = \sqrt[4]{\frac{4}{6} \cdot \frac{3}{5} \cdot \frac{2}{4} \cdot \frac{1}{3}}$$

- Brevity penalty:

$$BP = \min(1, 6/5)$$

Today

Natural language understanding

Two tasks (intent detection and slot filling): identify speaker's intent and extract semantic constituents from the natural language query

Sentence	first	class	fares	from	boston	to	denver
Slots	B-class_type	I-class_type	O	O	B-fromloc	O	B-toloc
Intent	airfare						

Figure: ATIS corpus sample with intent and slot annotation

- Intent detection is a classification task
- Slot filling is a sequence labelling task

NLU datasets: ATIS [1], Snips [2]

Joint intent detection and slot filling [3]

1. The encoder models is a biLSTM
2. The decoder is a unidirectional LSTM
3. At each step the decoder state s_i is: $s_i = f(s_{i-1}, y_{i-1}, h_i, c_i)$, where $c_i = \sum_j^T \alpha_{i,j} h_j$,
$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_k^T \exp(e_{i,k})},$$
$$e_{i,k} = g(s_{i-1}, h_k)$$

The inputs are explicitly aligned. Costs from both decoders are back-propagated to the encoder.

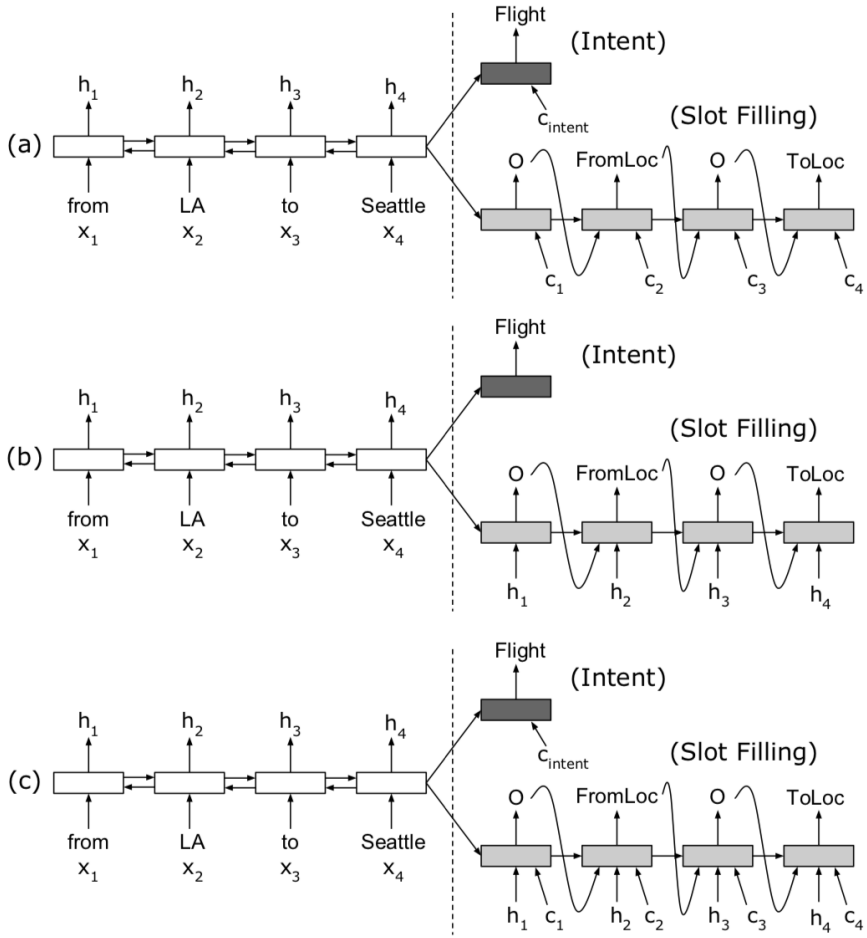


Figure: Encoder-decoder models

Joint intent detection and slot filling [3]

- BiLSTM reads the source sequence
- forward RNN models slot label dependencies
- the hidden state h_i at each step is a concatenation of the forward state fh_i and backward state bh_i
- the hidden state is h_i combined with the context vector c_i
- c_i is calculated as a weighted average of $h = (h_1, ..., h_T)$

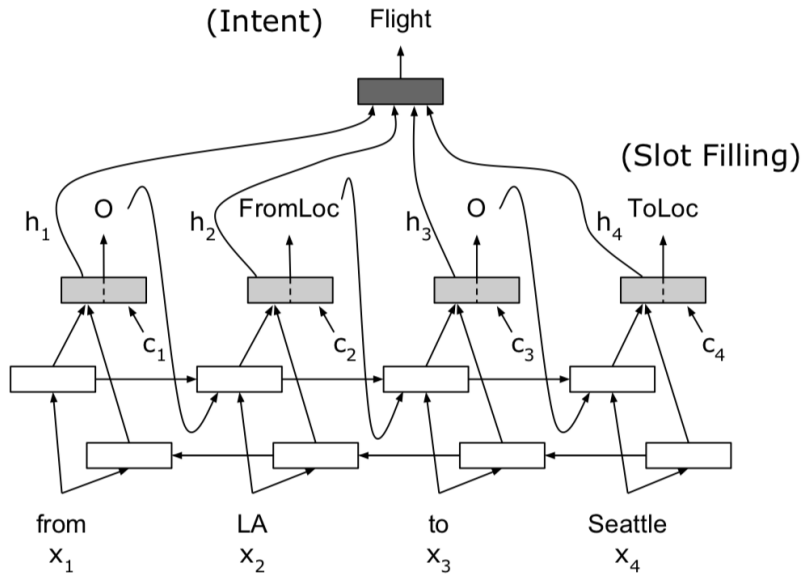


Figure: RNN-based model

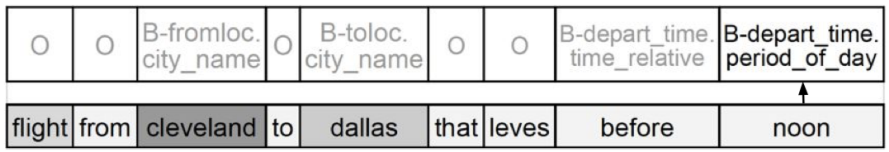


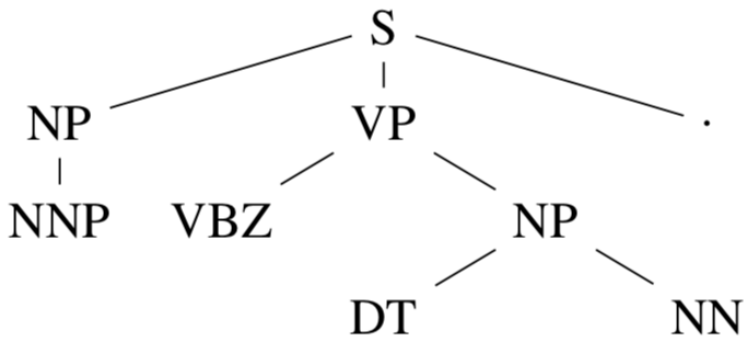
Figure: Attention weights

Today

1. Machine translation
2. Task oriented chat-bots
3. Constituency parsing
4. Spelling correction
5. Summarization
6. Question answering
7. IR-based QA
   - Datasets
   - Models

Grammar as a Foreign Language [4]

John has a dog . →



John has a dog . → (S (NP NNP)$_{\text{NP}}$ (VP VBZ (NP DT NN)$_{\text{NP}}$)$_{\text{VP}}$.)$_{\text{S}}$

Figure: Example parsing task and its linearization

Grammar as a Foreign Language [4]

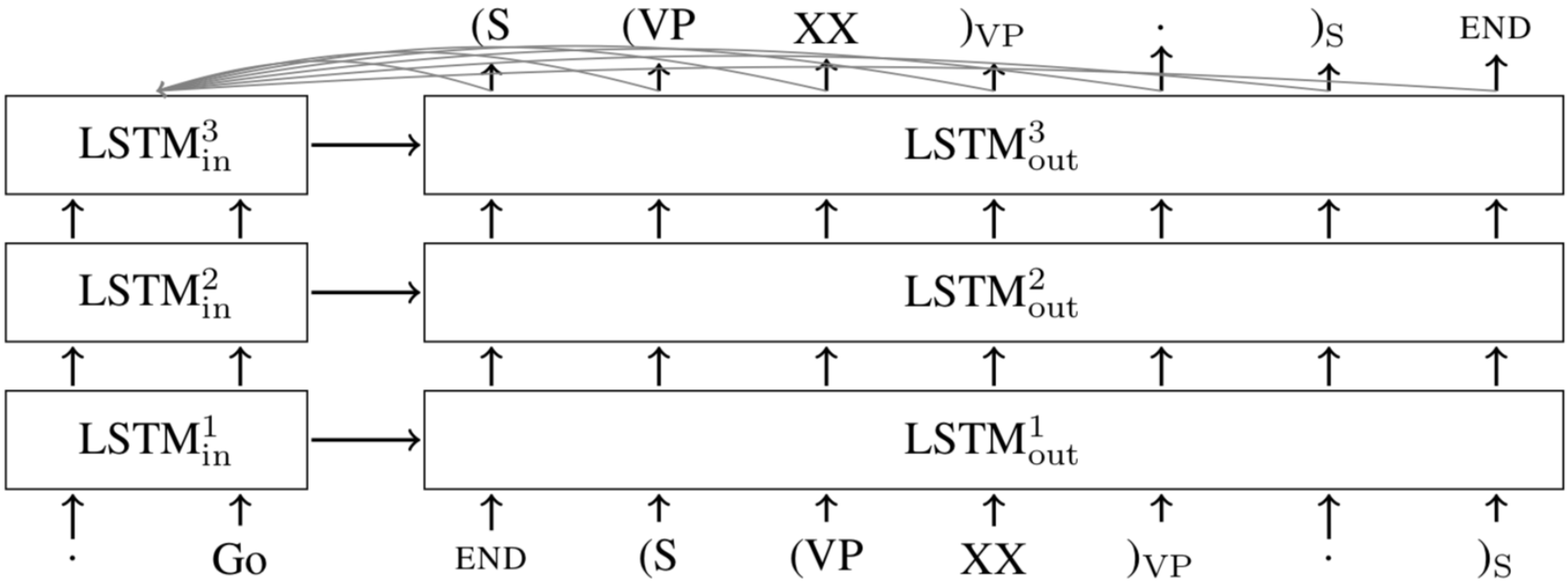


Figure: LSTM+attention encoder-decoder model for parsing

Grammar as a Foreign Language [4]

- The encoder LSTM is is used to encode the sequence of input words $A_i, |A| = T_A$
- The decoder LSTM is used to output symbols $B_i, |B| = T_B$
- The attention vector at each output time t over the input words:

$$u_i^t = v^T \tanh(W_1 h_i^E + W_2 h_t^D)$$

$$a_i^t = \text{softmax}(u_i^T)$$

$$d_t' = \sum_{i=1}^{T_A} a_i^t h_i^E,$$

where the vector v and matrices W_1, W_2 are learnable parameters of the model.

Today

Neural Language Correction with Character-Based Attention [5]

- Trained on a parallel corpus of "bad" (x) and "good" (y) sentences
- Encoder has a pyramid structure:

$$f_t^{(j)} = \text{GRU}(f_{t-1}^{(j-1)}, c_t^{(j-1)})$$
$$b_t^{(j)} = \text{GRU}(b_{t+1}^{(j-1)}, c_t^{(j-1)})$$
$$h_t^{(j)} = f_t^{(j)} + b_t^{(j)}$$
$$c_t^{(j)} = \tanh(W_{pyr}^{(j)}[h_{2t}^{(j-1)}, h_{2t+1}^{(j-1)}]^\top + b_{pyr}^{(j)})$$

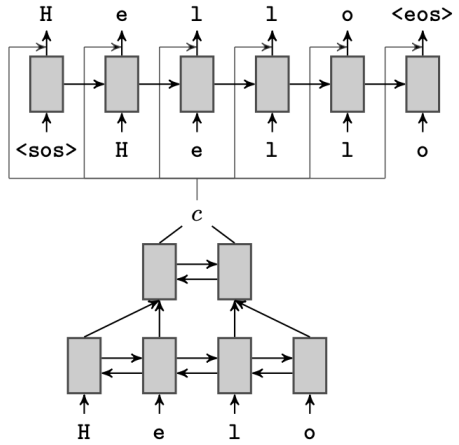


Figure: An encoder-decoder neural network model with two encoder hidden layers and one decoder hidden layer

Neural Language Correction with Character-Based Attention [5]

- Decoder network:
  $d_t^{(j)} = \text{GRU}(d_{t-1}^{(j-1)}, {}^{(j-1)}_{,t})$
- Attention mechanism:
  $u_{tk} = \phi_1(d^{(M)})^\top \phi_2(c_k), \phi : tahn(W \times \cdot)$
  $\alpha_{tk} = \frac{u_{tk}}{\sum_j u_{tj}}$
  $a_t = \sum_j \alpha_{tj} c_j$
- Loss:
  $L(x, y) = \sum_{t=1}^{T} logP(y_t | x, y_{<t})$

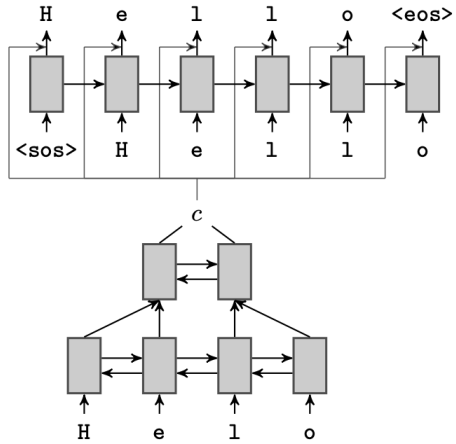


Figure: An encoder-decoder neural network model with two encoder hidden layers and one decoder hidden layer

Neural Language Correction with Character-Based Attention [5]

- Beam search for decoding:
  $s_k(y_{1:k}|x) =$
  $\log P_{NN}(y_{1:k}|x) + \lambda log P_{LM}(y_{1:k})$
- Synthezing errors: article or determiner errors (ArtOrDet) and noun number errors (Nn)

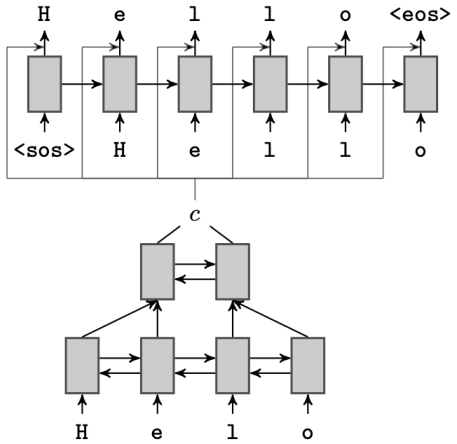


Figure: An encoder-decoder neural network model with two encoder hidden layers and one decoder hidden layer

Today

1. Machine translation
2. Task oriented chat-bots
3. Constituency parsing
4. Spelling correction
5. Summarization
6. Question answering
7. IR-based QA
   - Datasets
   - Models

Summarization & Simplification

Summarization

Summarization is the task of producing a shorter version of one or several documents that preserves most of the input's meaning.

1. **Abstractive** summarization: paraphrase the corpus using novel sentences
2. **Extractive** summarization: concatenate extracts taken from a corpus into a summary

Simplification

Simplification consists of modifying the content and structure of a text in order to make it easier to read and understand, while preserving its main idea and approximating its original meaning.

Image source: nlpprogress.com

Metrics: ROUGE [6]

Recall-Oriented Understudy for Gisting Evaluation

ROUGE is used to compare a system summary or translation against a set of reference human summaries:

$$\texttt{ROUGE}_n = \frac{\textit{number of overlapping n-grams}}{\textit{number of n-grams in reference summary}}$$

$$R_{LCS} = \frac{LCS(X,Y)}{|X|}, P_{LCS} = \frac{LCS(X,Y)}{|Y|}, \texttt{ROUGE}_L = \frac{(1+\beta^2)R_{LCS}P_{LCS}}{R_{LCS}+\beta^2 P_{LCS}},$$

where $LCS(X,Y)$ is the length of a longest common subsequence of X and Y.

Metrics: METEOR [7]

Metric for Evaluation of Translation with Explicit ORdering

METEOR is used to compare a system summary or translation against a set of reference human summaries:

$$\mathrm{P} = \frac{number\ of\ overlapping\ words}{number\ of\ words\ in\ system\ summary}$$

,

$$\mathrm{R} = \frac{number\ of\ overlapping\ words}{number\ of\ words\ in\ reference\ summary}$$

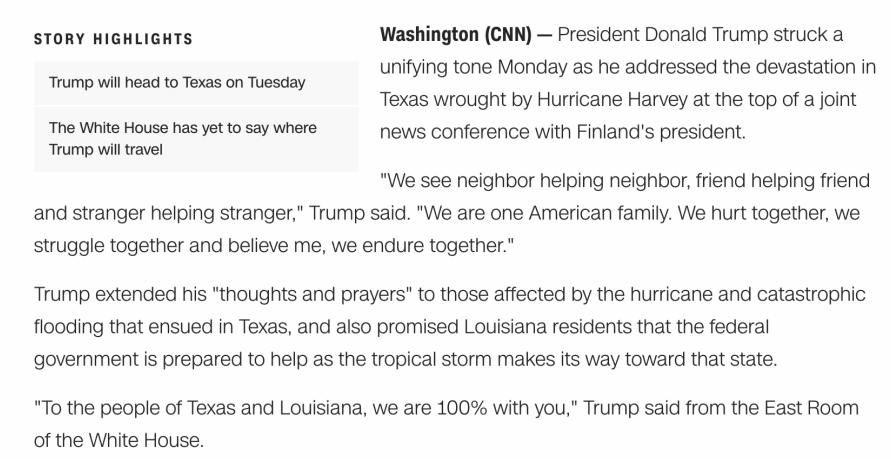
,

$$F_{mean} = \frac{10PR}{R + 9P}, penalty = 0.5(\frac{number\ of\ chunks}{number\ of\ overlapping\ words})^3$$

$$\mathrm{M} = F_{mean}(1 - p)$$

Datasets: CNN / Daily Mail [8], [9]

The dataset contains online news articles (781 tokens on average) paired with multi-sentence summaries (3.75 sentences or 56 tokens on average). The processed version contains 287,226 training pairs, 13,368 validation pairs and 11,490 test pairs.



STORY HIGHLIGHTS

Trump will head to Texas on Tuesday

The White House has yet to say where Trump will travel

Washington (CNN) — President Donald Trump struck a unifying tone Monday as he addressed the devastation in Texas wrought by Hurricane Harvey at the top of a joint news conference with Finland's president.

"We see neighbor helping neighbor, friend helping friend and stranger helping stranger," Trump said. "We are one American family. We hurt together, we struggle together and believe me, we endure together."

Trump extended his "thoughts and prayers" to those affected by the hurricane and catastrophic flooding that ensued in Texas, and also promised Louisiana residents that the federal government is prepared to help as the tropical storm makes its way toward that state.

"To the people of Texas and Louisiana, we are 100% with you," Trump said from the East Room of the White House.

Datasets: Webis-TLDR-17 [10]

The dataset contains 4 million content-summary pairs from Reddit.

Example Submission

Title: Ultimate travel kit
Body: Doing some traveling this year and I am looking to build the ultimate travel kit ... So far I have a Bonavita 0.5L travel kettle and AeroPress. Looking for a grinder that would maybe fit into the AeroPress. This way I can stack them in each other and have a compact travel kit.
TL;DR: What grinder would you recommend that fits in AeroPress?

Example Comment (to a different submission)

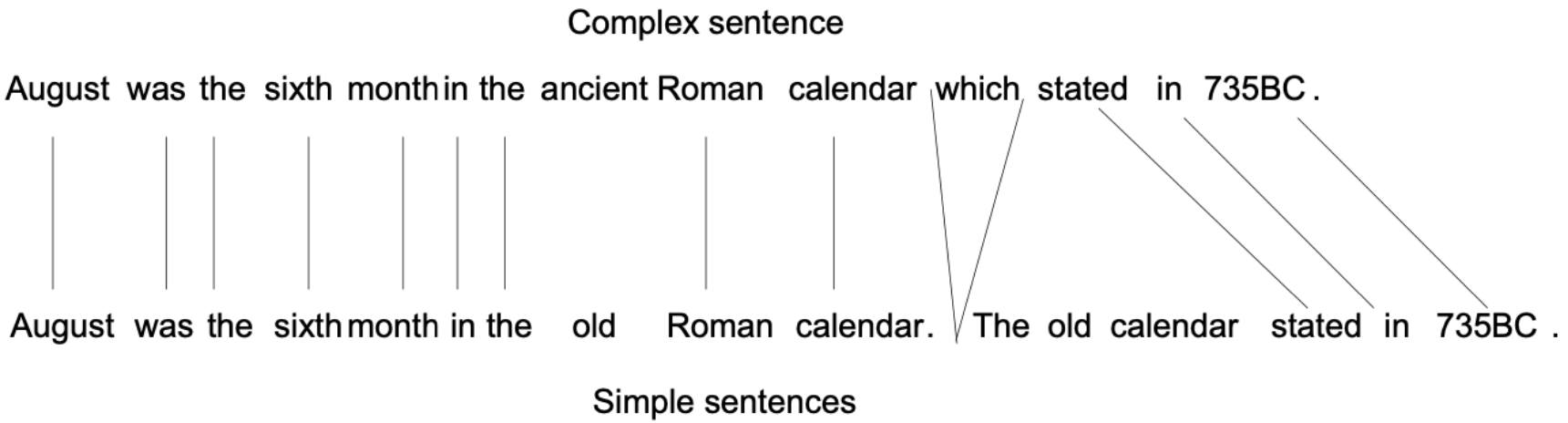
Body: Oh man this brings back memories. When I was little, around five, we were putting in a new shower system in the bathroom and had to open up the wall. The plumber opened up the wall first, then put in the shower system, and then left it there while he took a lunch break. After his break he patched up the wall and left, having completed the job. Then we couldn't find our cat. But we heard the cat. Before long we realized it was stuck in the wall, and could not get out. We called up the plumber again and he came back the next day and opened the wall. Out came our black cat, Socrates, covered in dust and filth.
TL;DR: plumber opens wall, cat climbs in, plumber closes wall, fucking meows everywhere until plumber returns the next day

Datasets: headline generation

1. Gigaword summarization dataset [11]
2. RIA news dataset [12]

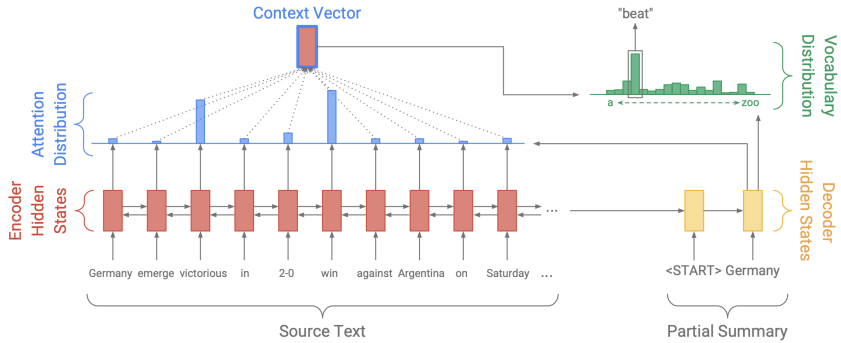
Datasets: WikiSmall [13]

Main source for simplified sentences is Simple English Wikipedia.
WikiSmall is a parallel corpus with more than 108K sentence pairs from 65,133 Wikipedia articles, allowing 1-to-1 and 1-to-N alignments



Get to the Point [14]

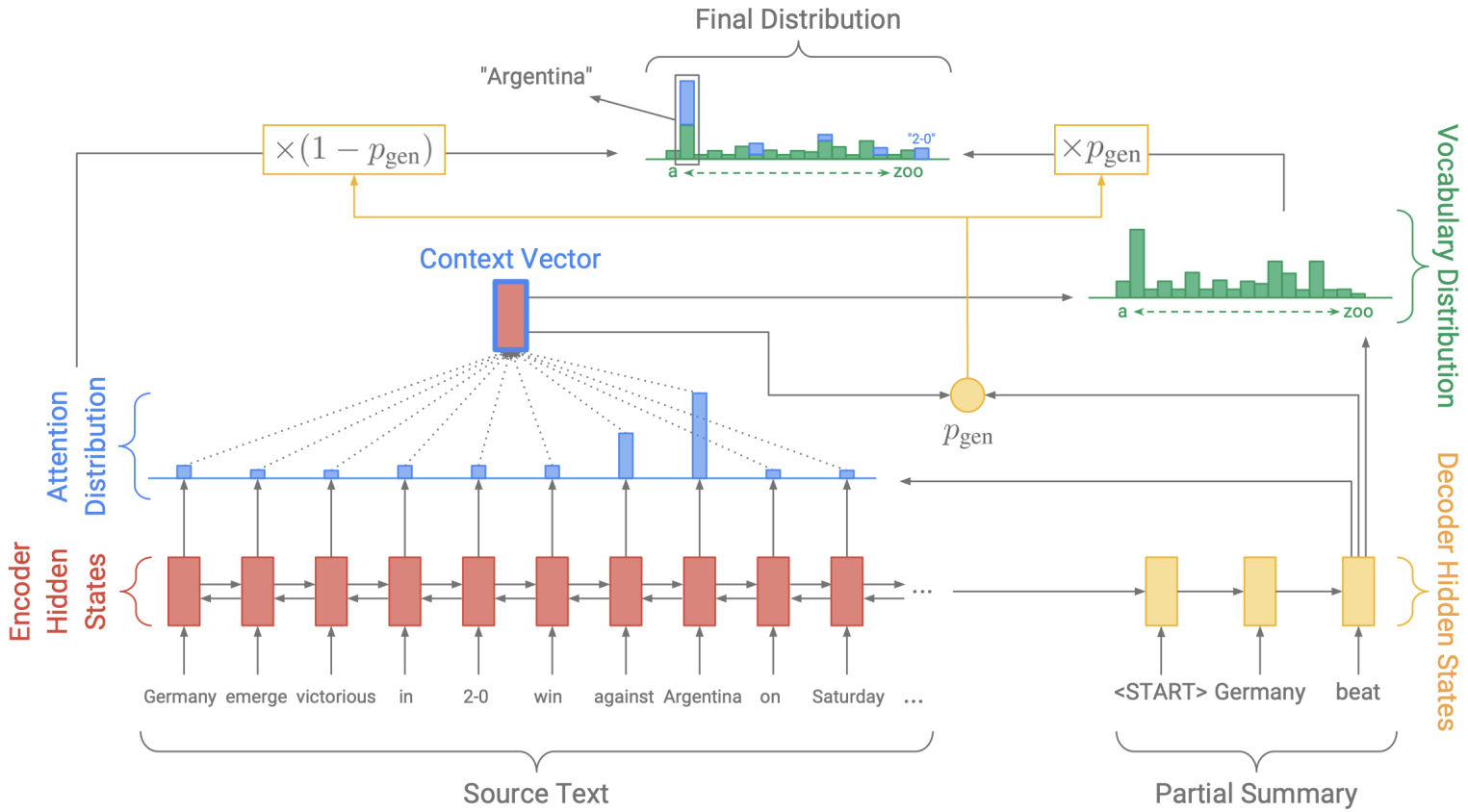
Sequence-to-sequence attentional model



- Bahdanau attention: $e_i^t = v^T \tanh(W_h h_i + W_s s_t + b_{attn})$, $a_t = \text{softmax}(e_t)$
- Context vector: $h_t = \sum_i a_i^t h_i$
- Vocabulary distribution: $P_{vocab} = \text{softmax}(V'(V[s_t, h_t] + b) + b)$
- NLL loss: $-\frac{1}{T} \sum_{t=0}^{T} \log P(w_t^*)$

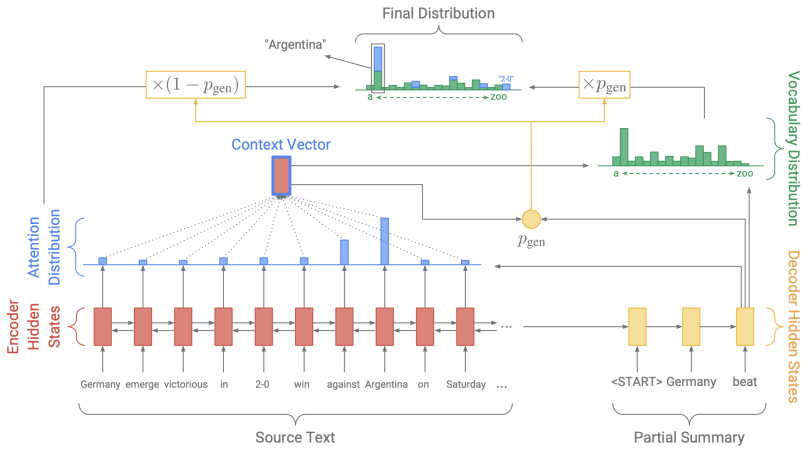
Get to the Point [14]

Pointer-generator model



Get to the Point [14]

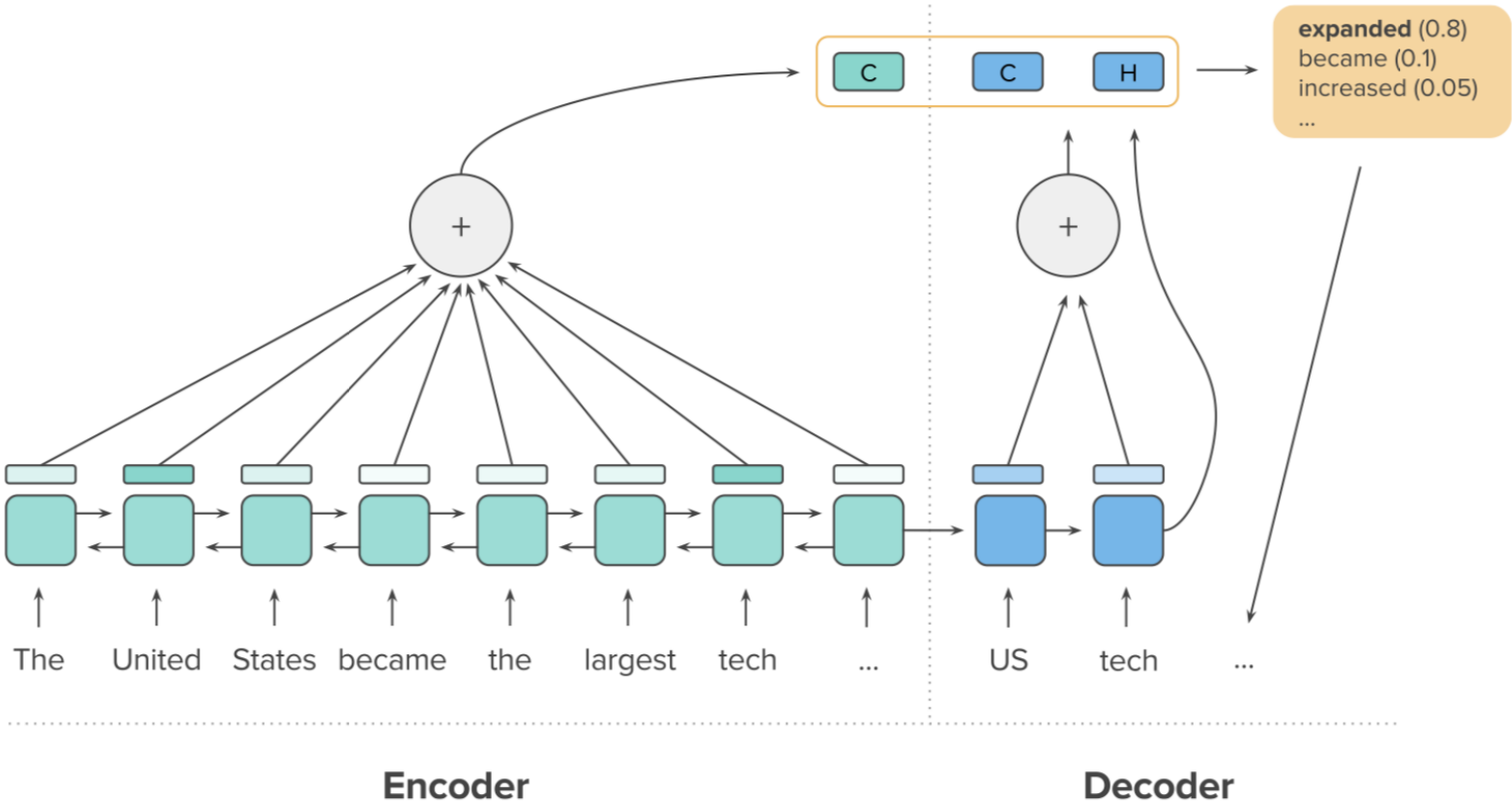
Pointer-generator model



- Generation probability: $p_{gen} = \sigma(w_{h^*}^T h_t + w_s^T s_t + w_x^T x_t + b_{ptr})$
- p_{gen} is used to switch between sampling from P_{vocab} or copying by sampling a^t
- $\mathrm{P(w)} = p_{gen} P_{vocab}(w) + (1 - p_{gen}) \sum_{i:w_i=w} a_i^t$

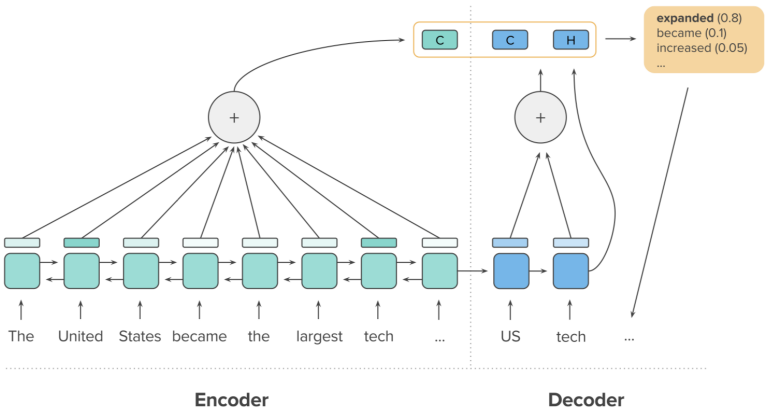
A Deep Reinforced Model

for Abstractive Summarization [15]



A Deep Reinforced Model

for Abstractive Summarization [15]



- Intra-temporal attention:
  $e_{ti} = h_t^{d^T} W_{attn}^e h_i^e$,
  $\alpha_{ti}^e = \text{softmax}(e_{ti})$,
  $c_t^e = \sum_{i=1}^{n} \alpha_{ti}^e h_i^e$
- Intra-decoder attention:
  $e_{tt'} = h_t^{d^T} W_{attn}^d h_i^d$,
  $\alpha_{tt'}^d = \text{softmax}(e_{tt'})$,
  $c_t^d = \sum_{i=j}^{t-1} \alpha_{tj}^d h_k^d$

A Deep Reinforced Model

for Abstractive Summarization [15]

- Token generation:

$$p(y_t|u_t = 0) = \text{softmax}(W_{out}[h_t^d, c_t^e, c_t^d] + b_{out})$$

- Pointer:

$$p(y_t = x_i|u = 1) = \alpha_{ti}^e$$

$$p(u_t = 1) = \sigma(W_u[h_t^d, c_t^e, c_t^d] + b_u)$$

- Probability distribution for the output token:

$$p(y_t) = p(u_t = 1)p(y_t|u_t = 1) + p(u_t = 0)p(y_t|u)t = 0)$$

- Sharing decoder weights: $W_{out} = \tanh(W_{emb}W_{proj})$

A Deep Reinforced Model

for Abstractive Summarization [15]

- Hybrid learning objective:

$$L_{mixed} = \gamma L_{rl} + (1\gamma) L_{ml}$$

- Teacher forcing:

$$L_{ml} = \sum_{t=1}^{n'} \log p(y_t | y_1, \ldots, y_{t-1}, x)$$

- Policy learning:

$$L_{rl} = (r(\hat{y}) - r(y^s)) \sum_{t=1}^{n'} \log p(y_t^s | y_1^s, \ldots, y_{t-1}^s, x),$$

where r is a reward function, $\hat{y}$ is the baseline output, obtained by maximizing the output probability distribution at each time step.

Today

1. Machine translation
2. Task oriented chat-bots
3. Constituency parsing
4. Spelling correction
5. Summarization
6. Question answering
7. IR-based QA
   - Datasets
   - Models

Types of questions

1. Factoid questions:
   - What is the dress code for the Vatican?
   - Who is the President of the United States?
   - What are the dots in Hebrew called?
2. Commonsense questions:
   - What do all humans want to experience in their own home? (a) feel comfortable, (b) work hard, (c) fall in love, (d) lay eggs, (e) live forever
3. Opinion questions:
   - Can anyone recommend a good coffee shop near HSE campus?
4. Cloze-style questions

Types of questions

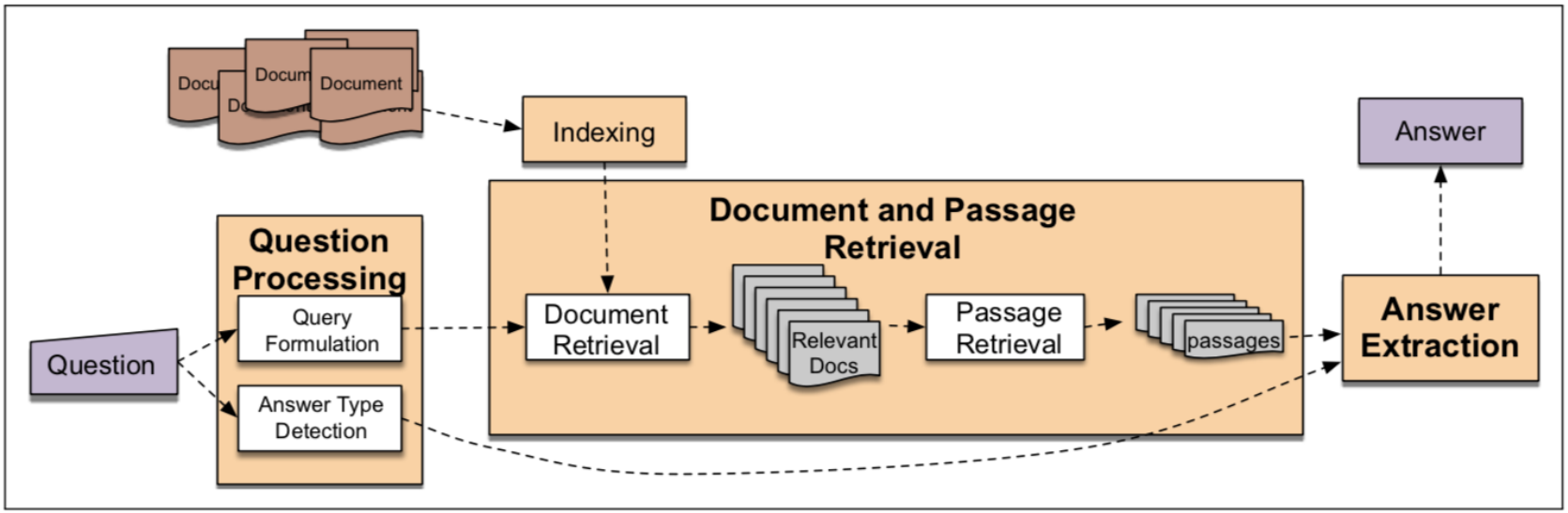
1. Types of answers
   - binary (yes / now)
   - find a span of text
   - multiple choice

Major paradigms for factoid question answering

1. Information retrieval (IR)-based QA: find a span of text, which answers a question
2. Open-domain Question Answering (ODQA): answer questions about nearly anything
3. Knowledge (KB)-based QA: build a semantic representation of question are used to question knowledge bases
   When Bernardo Bertolucci died? $\rightarrow$ `death-year(Bernardo Bertolucci, ?x)`

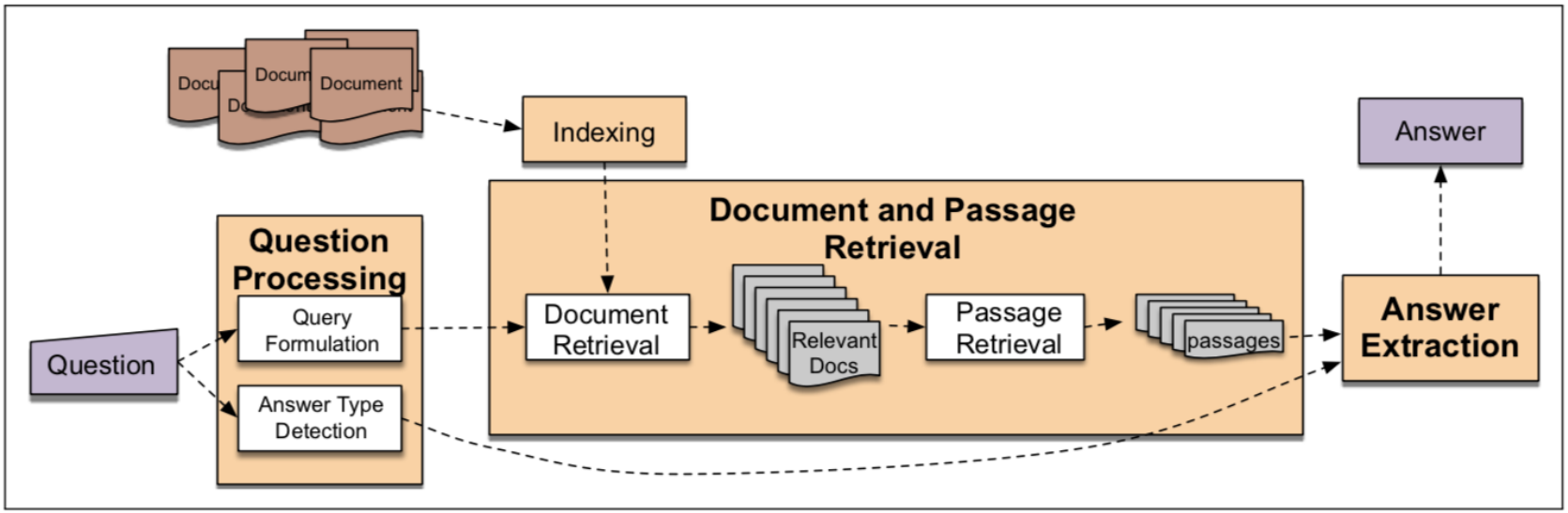
Today

IR-based QA



1. Question processing
   - answer type (PER, LOC, TIME)
   - focus
   - question type
2. Query formulation
   - question reformulation: remove *wh*-words, change word order
   - query expansion

IR-based QA



3. Document and passage retrieval
4. Answer extraction
   What are the dots in Hebrew called?
   In Hebrew orthography, **niqqud or nikkud**, *is a system of diacritical signs used to represent vowels or distinguish between alternative pronunciations of letters of the Hebrew alphabet.*

1. Machine translation
2. Task oriented chat-bots
3. Constituency parsing
4. Spelling correction
5. Summarization
6. Question answering
7. IR-based QA
   - Datasets
   - Models

Datasets for IR-based QA

Passage: Tesla later approached Morgan to ask for more funds to build a more powerful transmitter. **When asked where all the money had gone, Tesla responded by saying that he was affected by the Panic of 1901**, which he (Morgan) had caused. Morgan was shocked by the reminder of his part in the stock market crash and by Tesla's breach of contract by asking for more funds. Tesla wrote another plea to Morgan, but it was also fruitless. Morgan still owed Tesla money on the original agreement, and Tesla had been facing foreclosure even before construction of the tower began.

Question: On what did Tesla blame for the loss of the initial money?

Answer: Panic of 1901

1. Stanford Question Answering Dataset (SQuAD)
2. NewsQA
3. WikiQA
4. WebQuestions
5. WikiMovies
6. Russian: SberQUAD
7. MedQuAD [16]

SQuAD2.0 [17], [18]

100,000 questions in SQuAD1.1 and over 50,000 unanswerable questions in SQuAS2.0

1. Project Nayuki's Wikipedia's internal PageRanks to obtain the top 10000 articles of English Wikipedia, from which we sampled 536 articles uniformly at random
2. Articles splitted in individual paragraphes
3. Crowsourcing: ask and answer up to 5 questions on the content of that paragraph
4. Crowdworkers were encouraged to ask questions in their own words, without copying word phrases from the paragraph
5. Analysis: the (i) diversity of answer types, (ii) the difficulty of questions in terms of type of reasoning required to answer them, and (iii) the degree of syntactic divergence between the question and answer sentences.

`https://rajpurkar.github.io/SQuAD-explorer/`

RACE [19]

Passage:
In a small village in England about 150 years ago, a mail coach was standing on the street. It didn't come to that village often. People had to pay a lot to get a letter. The person who sent the letter didn't have to pay the postage, while the receiver had to.
"Here's a letter for Miss Alice Brown," said the mailman.
" I'm Alice Brown," a girl of about 18 said in a low voice.
Alice looked at the envelope for a minute, and then handed it back to the mailman.
"I'm sorry I can't take it, I don't have enough money to pay it", she said.
A gentleman standing around were very sorry for her. Then he came up and paid the postage for her.
When the gentleman gave the letter to her, she said with a smile, " Thank you very much, This letter is from Tom. I'm going to marry him. He went to London to look for work. I've waited a long time for this letter, but now I don't need it, there is nothing in it."
"Really? How do you know that?" the gentleman said in surprise.
"He told me that he would put some signs on the envelope. Look, sir, this cross in the corner means that he is well and this circle means he has found work. That's good news."
The gentleman was Sir Rowland Hill. He didn't forgot Alice and her letter.
"The postage to be paid by the receiver has to be changed," he said to himself and had a good plan.
"The postage has to be much lower, what about a penny? And the person who sends the letter pays the postage. He has to buy a stamp and put it on the envelope." he said . The government accepted his plan. Then the first stamp was put out in 1840. It was called the "Penny Black". It had a picture of the Queen on it.

Questions:

1): The first postage stamp was made _.
A. in England B. in America C. by Alice D. in 1910

2): The girl handed the letter back to the mailman because _ .
A. she didn't know whose letter it was
B. she had no money to pay the postage
C. she received the letter but she didn't want to open it
D. she had already known what was written in the letter

3): We can know from Alice's words that _ .
A. Tom had told her what the signs meant before leaving
B. Alice was clever and could guess the meaning of the signs
C. Alice had put the signs on the envelope herself
D. Tom had put the signs as Alice had told him to

4): The idea of using stamps was thought of by _ .
A. the government
B. Sir Rowland Hill
C. Alice Brown
D. Tom

5): From the passage we know the high postage made _ .
A. people never send each other letters
B. lovers almost lose every touch with each other
C. people try their best to avoid paying it
D. receivers refuse to pay the coming letters

Answer: ADABC

Figure: An example from RACE dataset

RACE consists of near 28k passages and near 100k questions generated by human experts (English instructors), and covers a variety of topics which are carefully designed for evaluating the students' ability in understanding and reasoning.

RACE [19]

Dataset	RACE-M	RACE-H	RACE	CNN	SQUAD	NEWSQA
Word Matching	29.4%	11.3%	15.8%	13.0%†	39.8%*	32.7%*
Paraphrasing	14.8%	20.6%	19.2%	41.0%†	34.3%*	27.0%*
Single-Sentence Reasoning	31.3%	34.1%	33.4%	19.0%†	8.6%*	13.2%*
Multi-Sentence Reasoning	22.6%	26.9%	25.8%	2.0%†	11.9%*	20.7%*
Ambiguous/Insufficient	1.8%	7.1%	5.8%	25.0%†	5.4%*	6.4%*

Figure: Statistic information about Reasoning type in different datasets

RACE includes five classes of questions: word matching, paraphrasing, single-sentence reasoning, multi-sentence reasoning, insufficient or ambiguous questions.

`http://www.cs.cmu.edu/~glai1/data/race/`

MS Marco

Field	Description
Query	A question query issued to Bing.
Passages	Top 10 passages from Web documents as retrieved by Bing. The passages are presented in ranked order to human editors. The passage that the editor uses to compose the answer is annotated as is_selected: 1.
Document URLs	URLs of the top ranked documents for the question from Bing. The passages are extracted from these documents.
Answer(s)	Answers composed by human editors for the question, automatically extracted passages and their corresponding documents.
Well Formed Answer(s)	Well-formed answer rewritten by human editors, and the original answer.
Segment	QA classification. E.g., tallest mountain in south america belongs to the ENTITY segment because the answer is an entity (Aconcagua).

Figure: The final dataset format for MS MARCO

Three tasks:

1. first predict whether a question can be answered, if so, generate the correct answer
2. the generated answer should be well-formed
3. the passage re-ranking

`http://www.msmarco.org`

DrQA [20]

Document Retriever: return 5 Wikipedia articles, using simple $tf - idf$-based retrieval

Document Reader: we are given a query $q = q_1, \ldots, q_l$ and n paragraphs $p_1, \ldots, p_m$

Question encoding: weighted sum of $RNN(q_1, \ldots, q_l)$

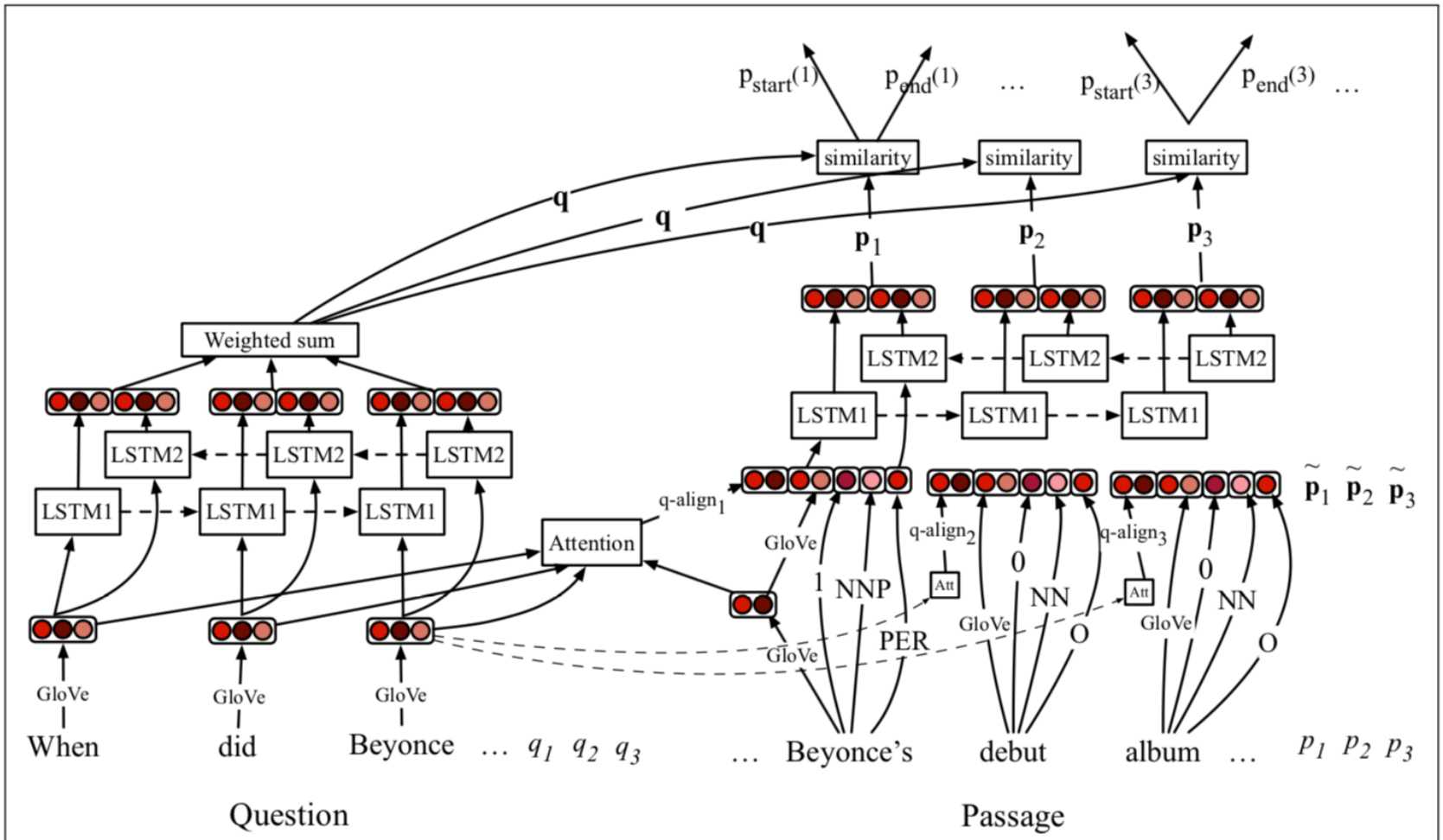
Paragraph encoding: $RNN(\widetilde{p_1}, \ldots, \widetilde{p_m})$, where $\widetilde{p_1}$ is comprised of:

- word embedding f_{emb}
- exact match $f_{exact\ match}$
- token features (POS, NER, TF), $f_{token\ features}$
- aligned question embedding $f_{align} = \sum_j a_{ij} q_j$

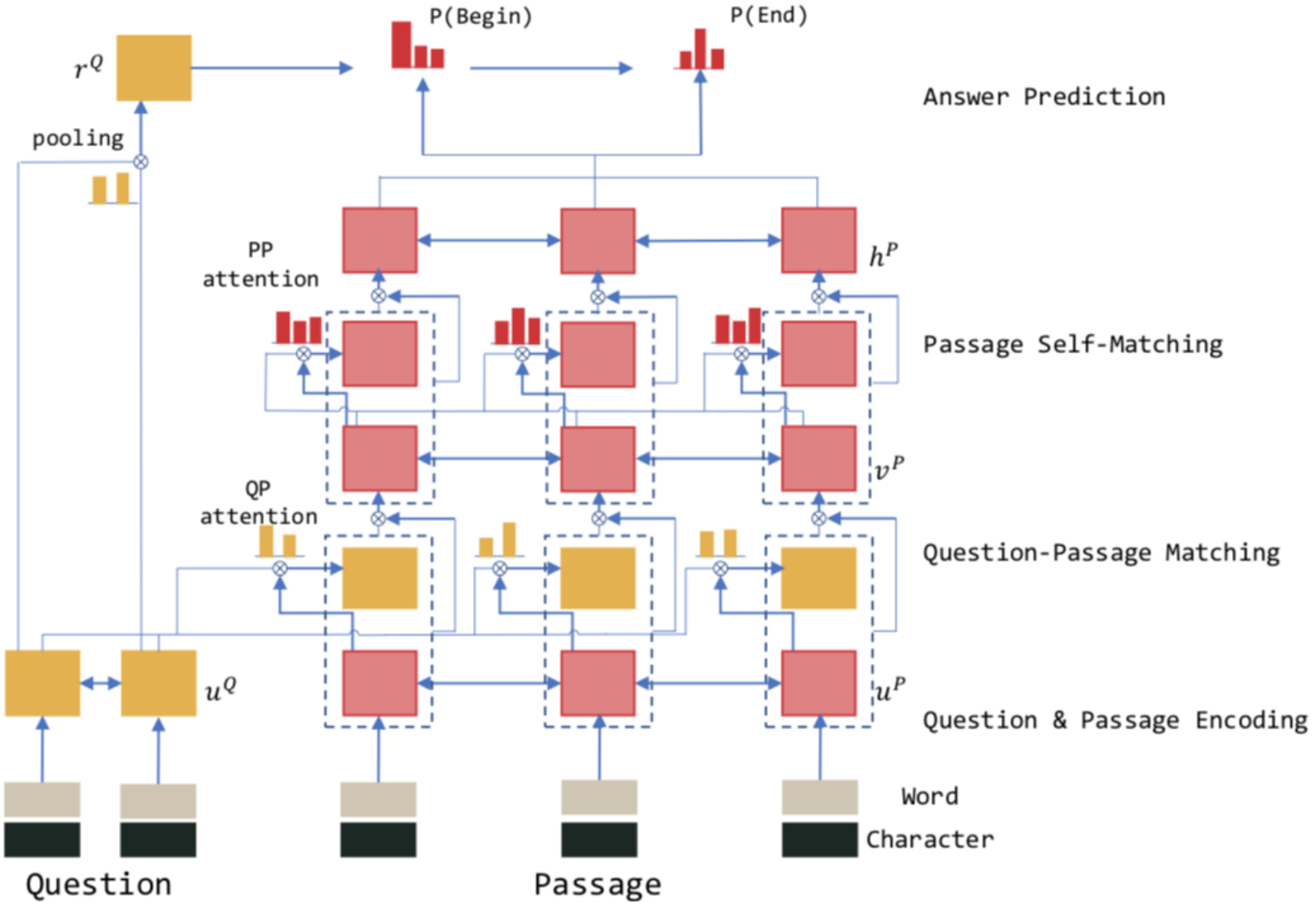
$$\frac{exp(\alpha(E(p_i)) \cdot exp(\alpha(E(q_i))}{\sum_{j'}(\alpha(E(p_i)) \cdot exp(\alpha(E(q_{j'}))}$$

DrQA [20]

Prediction: $P_{start} \propto exp(p_i W_s q)$, $P_{end} \propto exp(p_i W_e q)$
Choose the best span from token i to token i' such that $i \leq i' \leq i + 15$ and $P_{start}(i) \times P_{end}(i')$ is maximized.



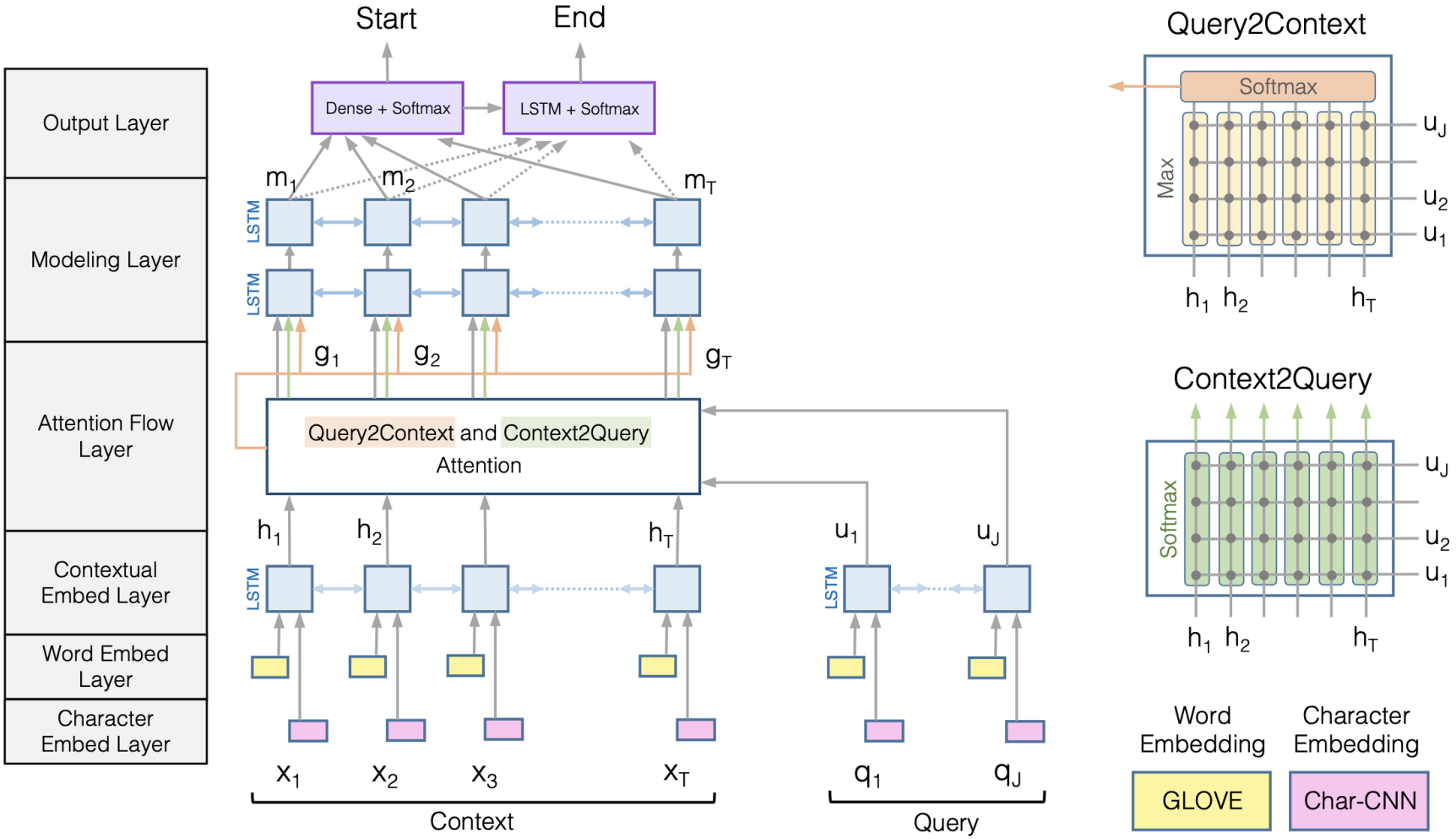
R-NET [21]


P(Begin)
P(End)
r^Q
pooling
Answer Prediction
PP attention
h^P
Passage Self-Matching
v^P
QP attention
Question-Passage Matching
u^Q
u^P
Question & Passage Encoding
Word
Character
Question
Passage

R-NET [21]

1. **Question and passage encoder**: BiRNN to convert the words to their respective word-level embeddings and and character-level embeddings
2. **Gated attention-based recurrent networks**: to incorporate question information into passage representation
3. **Self-matching attention**: passage context is necessary to infer the answer
4. **Output**: use pointer networks to predict the start and end position of the answer. To generate the initial hidden vector for the pointer network an attention-pooling over the question representation is used
5. **Training**: minimize the sum of the negative log probabilities of the ground truth start and end position by the predicted distributions

BiDAF [22]


Output Layer
Modeling Layer
Attention Flow Layer
Contextual Embed Layer
Word Embed Layer
Character Embed Layer
Start
End
Dense + Softmax
LSTM + Softmax
LSTM
Query2Context and Context2Query Attention
Context
Query
Query2Context
Softmax
Max
Context2Query
Softmax
Word Embedding
Character Embedding
GLOVE
Char-CNN

BiDAF [22]

1. **Character Embedding Layer** maps each word to a vector space using character-level CNNs
2. **Word Embedding Layer** maps each word to a vector space using a pre-trained word embedding model
3. **Contextual Embedding Layer** utilizes contextual cues from surrounding words to refine the embedding of the words. These first three layers are applied to both the query and context
4. **Attention Flow Layer** couples the query and context vectors and produces a set of query- aware feature vectors for each word in the context
5. **Modeling Layer employs** a Recurrent Neural Network to scan the context. 6. Output Layer provides an answer to the query.

Next generation of QA models

1. S-NET [23]: Extraction-then-synthesis framework
2. QANet [24] benefits from data augmentation techniques, such as paraphrasing and back translation
3. V-NET [25]: end-to-end neural model that enables answer candidates from different passages to verify each other based on their content representations
4. Deep Cascade QA [26]: deep cascade model, which consists of the document retrieval, paragraph retrieval and answer extraction modules

Take away messages

1. seq2seq architectures are exploited in a variety of NLP tasks
2. Attention mechanism helps to find soft alignments
3. The metrics are rarely differentiable, hence reinforcement learning

Reference I

C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, "The atis spoken language systems pilot corpus," in *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27, 1990*, 1990.

A. Coucke, A. Saade, A. Ball, T. Bluche, A. Caulier, D. Leroy, C. Doumouro, T. Gisselbrecht, F. Caltagirone, T. Lavril, M. Primet, and J. Dureau, "Snips voice platform: An embedded spoken language understanding system for private-by-design voice interfaces," *CoRR*, vol. abs/1805.10190, 2018. arXiv: 1805.10190. [Online]. Available: http://arxiv.org/abs/1805.10190.

B. Liu and I. Lane, "Attention-based recurrent neural network models for joint intent detection and slot filling," *arXiv preprint arXiv:1609.01454*, 2016.

Reference II


O. Vinyals, Ł. Kaiser, T. Koo, S. Petrov, I. Sutskever, and G. Hinton, "Grammar as a foreign language," in *Advances in neural information processing systems*, 2015, pp. 2773–2781.


Z. Xie, A. Avati, N. Arivazhagan, D. Jurafsky, and A. Y. Ng, *Neural language correction with character-based attention*, 2016. arXiv: 1603.09727 [cs.CL].


C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.


S. Banerjee and A. Lavie, "Meteor: An automatic metric for mt evaluation with improved correlation with human judgments," in *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005, pp. 65–72.

Reference III

K. M. Hermann, T. Kočiský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, *Teaching machines to read and comprehend*, 2015. arXiv: 1506.03340 [cs.CL].

R. Nallapati, B. Zhou, C. dos Santos, Ç. Gulcehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond," in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 280–290. DOI: 10.18653/v1/K16-1028. [Online]. Available: https://www.aclweb.org/anthology/K16-1028.

Reference IV

M. Völske, M. Potthast, S. Syed, and B. Stein, "TL;DR: Mining Reddit to learn automatic summarization," in *Proceedings of the Workshop on New Frontiers in Summarization*, Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 59–63. DOI: 10.18653/v1/W17-4508. [Online]. Available: https://www.aclweb.org/anthology/W17-4508.

A. M. Rush, S. Chopra, and J. Weston, *A neural attention model for abstractive sentence summarization*, 2015. arXiv: 1509.00685 [cs.CL].

D. Gavrilov, P. Kalaidin, and V. Malykh, *Self-attentive model for headline generation*, 2019. arXiv: 1901.07786 [cs.CL].

Reference V

Z. Zhu, D. Bernhard, and I. Gurevych, "A monolingual tree-based translation model for sentence simplification," in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, Beijing, China: Coling 2010 Organizing Committee, Aug. 2010, pp. 1353–1361. [Online]. Available: https://www.aclweb.org/anthology/C10-1152.

A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 1073–1083. DOI: 10.18653/v1/P17-1099. [Online]. Available: https://www.aclweb.org/anthology/P17-1099.

R. Paulus, C. Xiong, and R. Socher, *A deep reinforced model for abstractive summarization*, 2017. arXiv: 1705.04304 [cs.CL].

Reference VI


A. Ben Abacha and D. Demner-Fushman, "A question-entailment approach to question answering," *arXiv e-prints*, Jan. 2019. arXiv: 1901.08079 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1901.08079.


P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.


P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for squad," *arXiv preprint arXiv:1806.03822*, 2018.


G. Lai, Q. Xie, H. Liu, Y. Yang, and E. Hovy, "Race: Large-scale reading comprehension dataset from examinations," *arXiv preprint arXiv:1704.04683*, 2017.

Reference VII


D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading wikipedia to answer open-domain questions," *arXiv preprint arXiv:1704.00051*, 2017.


W. Wang, N. Yang, F. Wei, B. Chang, and M. Zhou, "Gated self-matching networks for reading comprehension and question answering," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2017, pp. 189–198.


M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," , 2016.


C. Tan, F. Wei, N. Yang, B. Du, W. Lv, and M. Zhou, "S-net: From answer extraction to answer synthesis for machine reading comprehension.," in *AAAI*, 2018.

Reference VIII

A. W. Yu, D. Dohan, M.-T. Luong, R. Zhao, K. Chen, M. Norouzi, and Q. V. Le, "Qanet: Combining local convolution with global self-attention for reading comprehension," *arXiv preprint arXiv:1804.09541*, 2018.

Y. Wang, K. Liu, J. Liu, W. He, Y. Lyu, H. Wu, S. Li, and H. Wang, "Multi-passage machine reading comprehension with cross-passage answer verification," *arXiv preprint arXiv:1805.02220*, 2018.

M. Yan, J. Xia, C. Wu, B. Bi, Z. Zhao, J. Zhang, L. Si, R. Wang, W. Wang, and H. Chen, "A deep cascade model for multi-document reading comprehension," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 7354–7361, Jul. 2019, ISSN: 2159-5399. DOI: 10.1609/aaai.v33i01.33017354. [Online]. Available: http://dx.doi.org/10.1609/aaai.v33i01.33017354.