

# Natural language processing

## 10. Question Answering

Ekaterina Chernyak

Faculty of Computer Science

November 28, 2018

## Intro

## IR-based QA

Datasets

Models

## KB QA

# Major paradigms for factoid question answering

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?
1. Information retrieval (IR)-based QA: find a span of text, which answers a question (reading comprehension)
  2. Knowledge (KB)-based QA: build a semantic representation of question are used to question knowledge bases  
*When Bernardo Bertolucci died?* → death-year(Bernardo Bertolucci, ?x)

Intro

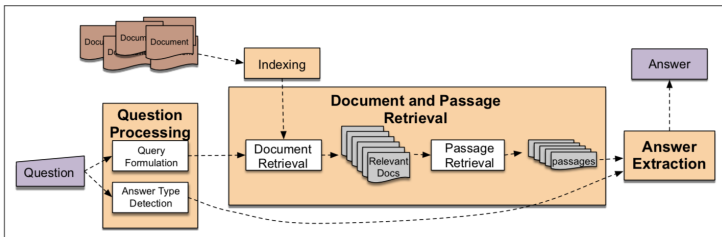
IR-based QA

Datasets

Models

KB QA

# IR-based QA



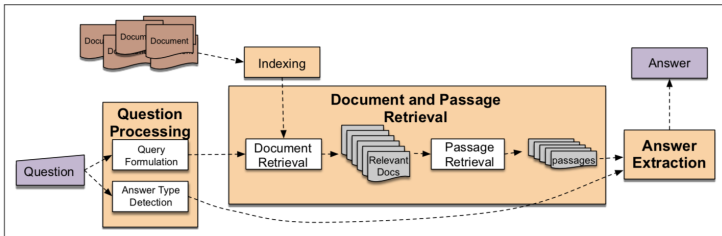
## 1. Question processing

- ▶ 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.*

Intro

IR-based QA

Datasets

Models

KB QA

# 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

---

Figure: An example from the SQuAD dataset

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



# SQuAD2.0 [RZLL16, RJL18]

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

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/>

Intro

IR-based QA

Datasets

Models

KB QA

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

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

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

**Paragraph encoding:**  $RNN(\tilde{p}_1, \dots, \tilde{p}_m)$ , where  $\tilde{p}_1$  is comprised of:

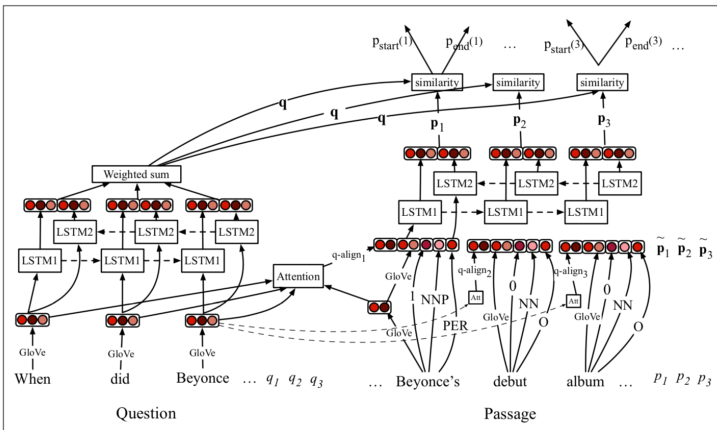
- ▶ word embedding  $f_{emb}$
- ▶ exact match  $f_{exact\ match}$
- ▶ token features (POS, NER, TF),  $f_{token\ features}$
- ▶ aligned question embedding  $f_{align} = \sum_j a_{ij} q_j$

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

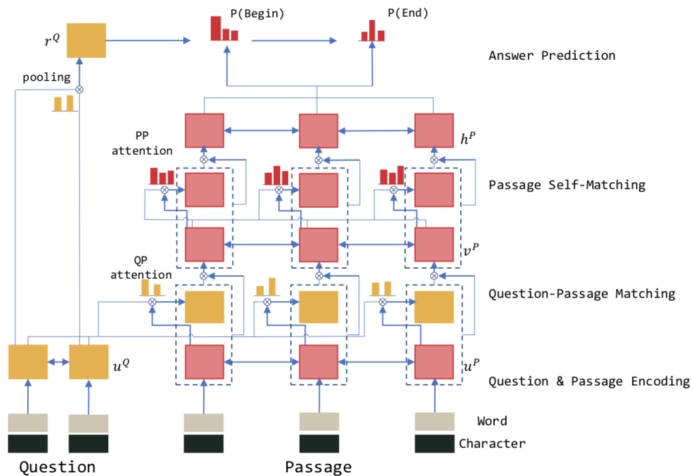
# DrQA [CFWB17]

**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[WYW<sup>+</sup>17]

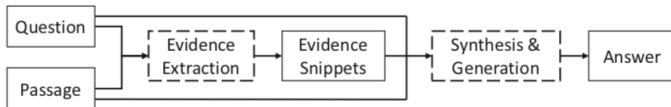


# R-NET[WYW<sup>+</sup>17]

1. **Question and passage encoder:** BiRNN to convert the words to their respective word-level embeddings 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

# S-NET [TWY<sup>+</sup>18]

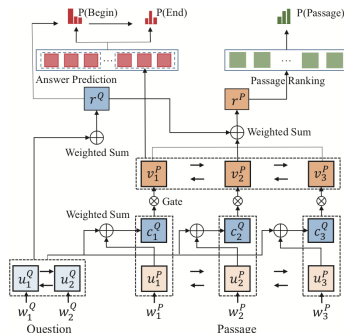
Extraction-then-synthesis framework



1. Evidence Extraction
2. Answer Synthesis

**Evidence Extraction:** is trained by minimizing joint objective functions of

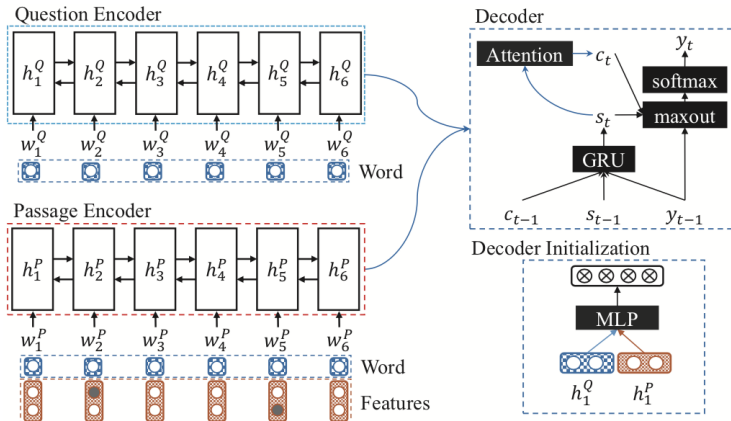
- ▶ **Evidence Snippet Prediction:** concatenate all passages to predict one span for the evidence snippet prediction using pointer networks
- ▶ **Passage Ranking:** match the question and each passage from word level to passage level





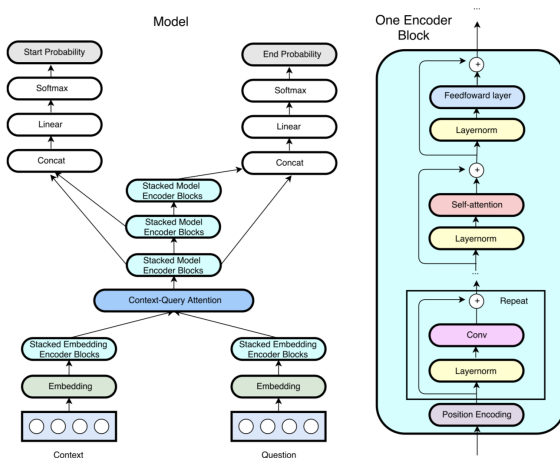
# S-NET [TWY<sup>+</sup>18]

**Answer Synthesis:** sequence-to-sequence model to synthesize the answer with the extracted evidences as features



# QANet [YDL<sup>+</sup>18]

Given a context paragraph with  $n$  words  $C = (c_1, c_2, \dots, c_n)$  and the query sentence with  $m$  words  $Q = (q_1, q_2, \dots, q_m)$ , output a span  $S = (c_i, c_{i+1}, \dots, c_{i+j})$  from the original paragraph  $C$ .



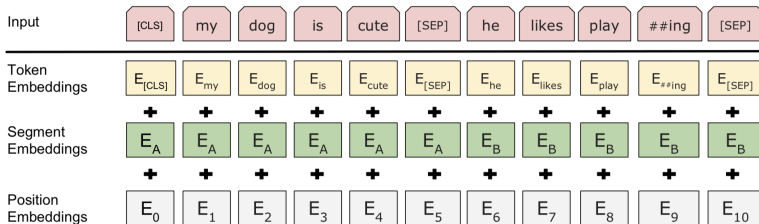
# QANet [YDL<sup>+</sup>18]

1. **Input Embedding Layer:** word embedding and character embedding
2. **Embedding Encoder Layer:** [convolution-layer + self-attention-layer + feed-forward-layer]
3. **Context-Query Attention Layer**
4. **Model Encoder Layer**
5. **Task-specific output layer:** predict the probability of each position in the context being the start or end of an answer span
6. Data Augmentation by backtranslation and paraphrasing

# BERT [DCLT18]

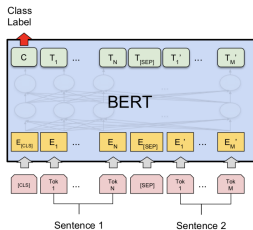
## Bidirectional Encoder Representations from Transformers

- ▶  $L$  – number of Transformer blocks,  $H$  – hidden size,  $A$  – the number of self-attention heads
- ▶ BERT<sub>BASE</sub>:  $L=12$ ,  $H=768$ ,  $A=12$ , Total Parameters=110M
- ▶ Embeddings: WordPiece + position + segment
- ▶ **Two tasks**: Masked LM, Next Sentence Prediction

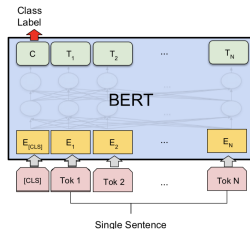


# BERT [DCLT18]

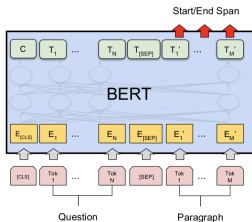
## Bidirectional Encoder Representations from Transformers



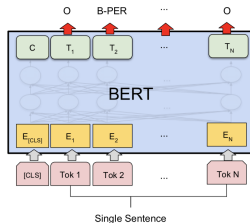
(a) Sentence Pair Classification Tasks:  
MNLI, QQP, QNLI, STS-B, MRPC,  
RTE, SWAG



(b) Single Sentence Classification Tasks:  
SST-2, CoLA



(c) Question Answering Tasks:  
SQuAD v1.1



(d) Single Sentence Tagging Tasks:  
CoNLL-2003 NER

# Models

- ▶ DrQA, R-NET, S-NET – deep RNNs
- ▶ QANet, BERT – CNNs + self-attention

Intro

IR-based QA

Datasets

Models

KB QA

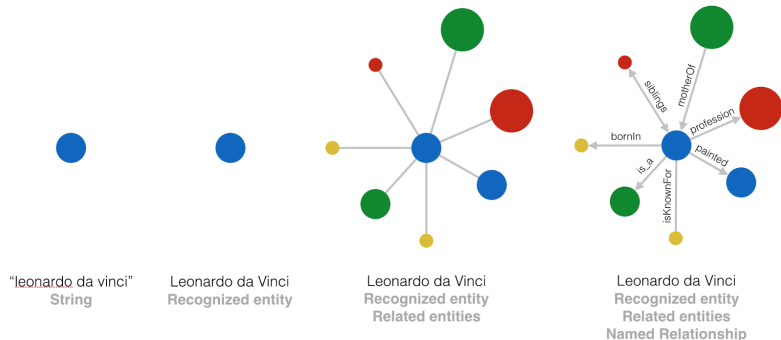
# Knowledge-based QA

subject	predicate	object
Lyubov Polishchuk	death-date	28 November 2006

- ▶ When Lyubov Polishchuk died?
  - ▶ Who died on 28 November 2006?
1. **Rule-based methods:** patterns that search for the question word and main verb
  2. **OpenIE:** map between the words in question and canonical relations
  3. **Knowledge base / knowledge graph:** match the words to concepts and relations in KB / KG



# Knowledge representation



[https://medium.com/@sderymail/  
challenges-of-knowledge-graph-part-1-d9ffe9e35214](https://medium.com/@sderymail/challenges-of-knowledge-graph-part-1-d9ffe9e35214)

# Datasets

What American cartoonist is the creator of Andy Lippincott?	(andy.lippincott, character.created.by, <u>garry.trudeau</u> )
Which forest is Fires Creek in?	(fires.creek, containedby, <u>nantahala.national.forest</u> )
What is an active ingredient in childrens earache relief ?	(childrens.earache.relief, active.ingredients, <u>capsicum</u> )
What does Jimmy Neutron do?	(jimmy.neutron, fictional.character.occupation, <u>inventor</u> )
What dietary restriction is incompatible with kimchi?	(kimchi, incompatible.with.dietary.restrictions, <u>veganism</u> )

**Figure:** Examples of simple QA extracted from the dataset SimpleQuestions. Actual answers are underlined.

- ▶ SimpleQuestions (100k questions) [BUCW15]: contains more than 100k questions written by human annotators and associated to Freebase facts,
- ▶ WebQuestions (6k questions) is created automatically using the Google suggest API.

## Memory Networks for Simple QA [BUCW15]

A Memory Network has four components: Input map (I), Generalization (G), Output map (O) and Response (R)

1. **Input:** stores and preprocesses Freebase facts, questions and Reverb facts are stored as BoW vectors ( $f(y) \in \mathbb{R}^{N_s}$ ,  $g(q) \in \mathbb{R}^{N_v}$ ,  $h(y) \in \mathbb{R}^{N_s+N_v}$ )
2. **Generalization:** adds new elements to the memory. The memory is a multigraph, each node is a Free-base entity and labeled arcs are Freebase relationships. Precomputed links are used to link Reverb facts to Freebase facts.
3. **Output:** performs the memory lookups given the input to return a single supporting fact.
  - ▶ **Candidate generation:** match  $n$ -grams to Freebase entities
  - ▶ **Scoring:**

$$S_{QA}(q, y) = \cos(W_V g(q), W_S f(y))$$

$$S_{RVB}(q, y) = \cos(W_V g(q), W_{VS} h(y))$$

4. **Response:** returns the set of objects of the selected supporting fact

Training: SGD on WARP loss with NS

# Embedding-based KB-QA [YDZR14]

1. Training triplet  $w = [\mathbb{C}, t, p]$ ,  $\mathbb{C} - n - \text{BoW}$  for question,  $t$  - entity types and  $p$  - predicates in Freebase. Each feature is encoded with an embedding.
2. Embeddings are trained under the soft ranking criterion, which conducts Stochastic Gradient Descent (SGD):





$$\forall i, \forall y' \neq y_i, \max(0, 1 - \text{Sim}(x_i, y_i) + \text{Sim}(x_i, y'))$$

# Quiz




Подробно ответьте на вопросы по ссылке. Ответы на вопросы принимаются до 23:59 28.11. Плагиат (т.е. заимствованные из интернета или у другого студента) приводит к обнулению оценки. Каждый вопрос оценивается 2 баллами.

<https://goo.gl/forms/uXUcr4MoY04qBs222>

# References I

-  Antoine Bordes, Nicolas Usunier, Sumit Chopra, and Jason Weston, *Large-scale simple question answering with memory networks*, arXiv preprint arXiv:1506.02075 (2015).
-  Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes, *Reading wikipedia to answer open-domain questions*, arXiv preprint arXiv:1704.00051 (2017).
-  Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint arXiv:1810.04805 (2018).
-  Pranav Rajpurkar, Robin Jia, and Percy Liang, *Know what you don't know: Unanswerable questions for squad*, arXiv preprint arXiv:1806.03822 (2018).

# References II

-  Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang, *Squad: 100,000+ questions for machine comprehension of text*, arXiv preprint arXiv:1606.05250 (2016).
-  Chuanqi Tan, Furu Wei, Nan Yang, Bowen Du, Weifeng Lv, and Ming Zhou, *S-net: From answer extraction to answer synthesis for machine reading comprehension.*, AAAI, 2018.
-  Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou, *Gated self-matching networks for reading comprehension and question answering*, Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, 2017, pp. 189–198.

# References III



Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le, *Qanet: Combining local convolution with global self-attention for reading comprehension*, arXiv preprint arXiv:1804.09541 (2018).



Min-Chul Yang, Nan Duan, Ming Zhou, and Hae-Chang Rim, *Joint relational embeddings for knowledge-based question answering*, Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 645–650.