School of CIT –
Social Computing
Research Group

Technical
University
of Munich

Natural Language Processing
IN2361

Prof. Dr. Georg Groh

Chapter 25
Question Answering

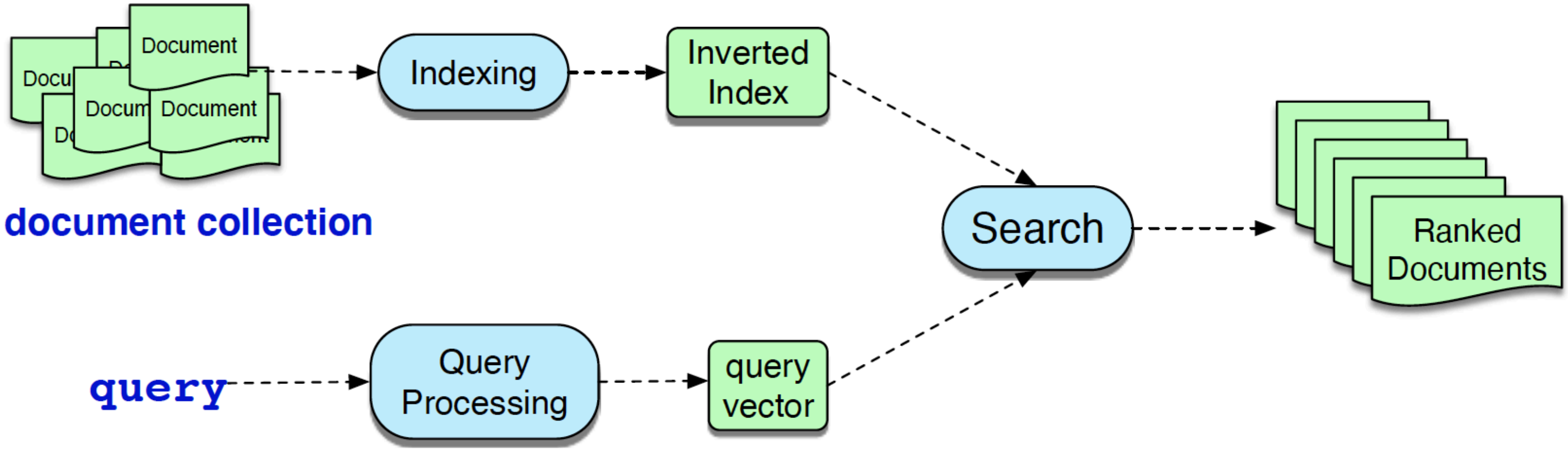
- content is based on [1]
- certain elements (e.g. equations or tables) were taken over or taken over in a modified form from [1]
- citations of [1] or from [1] are omitted for legibility
- errors are fully in the responsibility of Georg Groh
- BIG thanks to Dan and James for a great book!

IR-based Factoid Question Answering

- factoid question answering via Information Retrieval (i.e. by finding as answers short text segments on the Web or some other collection of documents)

Question	Answer
Where is the Louvre Museum located?	in Paris, France
What's the abbreviation for limited partnership?	L.P.
What are the names of Odin's ravens?	Huginn and Muninn
What currency is used in China?	the yuan
What kind of nuts are used in marzipan?	almonds
What instrument does Max Roach play?	drums
What's the official language of Algeria?	Arabic
How many pounds are there in a stone?	14

- Information Retrieval:



Information Retrieval

- Query-document matching: simple approach: tf-idf + cosine:

$$\mathrm{tf}_{t,d} = \log_{10}(\mathrm{count}(t,d)+1)$$

$$\mathrm{idf}_t = \log_{10}\frac{N}{\mathrm{df}_t}$$

$$\text{tf-idf}(t,d) = \mathrm{tf}_{t,d}\cdot \mathrm{idf}_t$$

$$\mathrm{score}(q,d) = \cos(\mathbf{q},\mathbf{d}) = \frac{\mathbf{q}\cdot\mathbf{d}}{|\mathbf{q}||\mathbf{d}|}$$

$$= \sum_{t\in\mathbf{q}} \frac{\text{tf-idf}(t,q)}{\sqrt{\sum_{q_i\in q}\text{tf-idf}^2(q_i,q)}}\cdot\frac{\text{tf-idf}(t,d)}{\sqrt{\sum_{d_i\in d}\text{tf-idf}^2(d_i,d)}}$$

$$\approx \sum_{t\in q}\frac{\text{tf-idf}(t,d)}{|d|}$$

Word	df	idf
Romeo	1	1.57
salad	2	1.27
Falstaff	4	0.967
forest	12	0.489
battle	21	0.246
wit	34	0.037
fool	36	0.012
good	37	0
sweet	37	0

- Query-document matching: simple approach: tf-idf + cosine:

Word	df	idf
Romeo	1	1.57
salad	2	1.27
Falstaff	4	0.967
forest	12	0.489
battle	21	0.246
wit	34	0.037
fool	36	0.012
good	37	0
sweet	37	0

In all Shakespeare plays

Query: sweet love

Doc 1: Sweet sweet nurse! Love?
Doc 2: Sweet sorrow
Doc 3: How sweet is love?
Doc 4: Nurse!

	Document 1					Document 2				
word	**count**	**tf**	**df**	**idf**	**tf-idf**	**count**	**tf**	**df**	**idf**	**tf-idf**
love	1	0.301	2	0.301	0.091	0	0	2	0.301	0
sweet	2	0.477	3	0.125	0.060	1	0.301	3	0.125	0.038
sorrow	0	0	1	0.602	0	1	0.301	1	0.602	0.181
how	0	0	1	0.602	0	0	0	1	0.602	0
nurse	1	0.301	2	0.301	0.091	0	0	2	0.301	0
is	0	0	1	0.602	0	0	0	1	0.602	0

$|d_1| = \sqrt{.091^2 + .060^2 + .091^2} = .141$ $\quad$ $|d_2| = \sqrt{.038^2 + .181^2} = .185$

Doc	$\|d\|$	**tf-idf**(*sweet*)	**tf-idf**(*love*)	**score**
1	.141	.060	.091	1.07
3	.274	.038	.091	0.471
2	.185	.038	0	0.205
4	.090	0	0	0

- Query-document matching: variant: BM25

$$\sum_{t \in q} \overbrace{\log\left(\frac{N}{df_t}\right)}^{\text{IDF}} \overbrace{\frac{tf_{t,d}}{k\left(1-b+b\left(\frac{|d|}{|d_{\text{avg}}|}\right)\right)+tf_{t,d}}}^{\text{weighted tf}}$$

k: adjust balance btw. tf and idf

b: control influence of document length normalization

$|d_{avg}|$: length of average document

- Query-document matching: variant: use stop word removal (controversial!)

IR: Inverted Index

- Inverted Index: dictionary+postings: hash-map of terms (keys) and list of references to documents containing term

Doc 1:	Sweet sweet nurse! Love?
Doc 2:	Sweet sorrow
Doc 3:	How sweet is love?
Doc 4:	Nurse!

how {1}	$\rightarrow$	3 [1]
is {1}	$\rightarrow$	3 [1]
love {2}	$\rightarrow$	1 [1] $\rightarrow$ 3 [1]
nurse {2}	$\rightarrow$	1 [1] $\rightarrow$ 4 [1]
sorry {1}	$\rightarrow$	2 [1]
sweet {3}	$\rightarrow$	1 [2] $\rightarrow$ 2 [1] $\rightarrow$ 3 [1]

document frequency

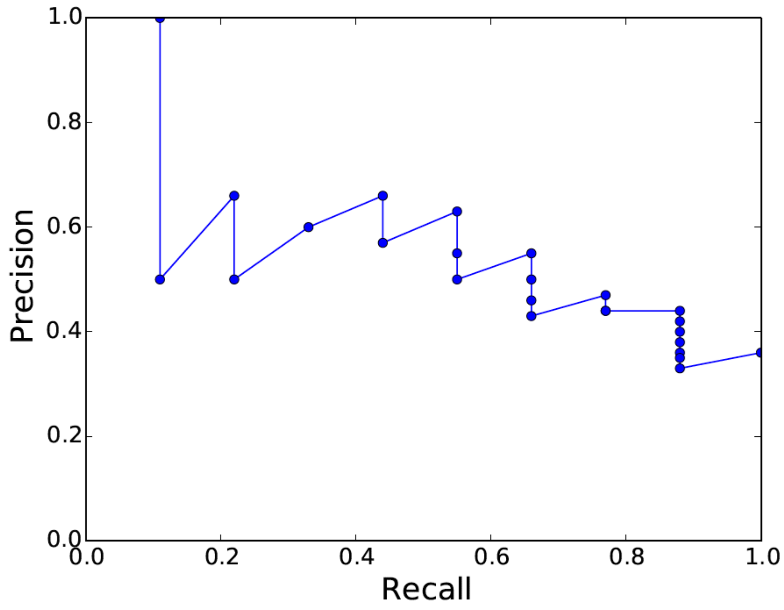
term frequency

IR: Performance Measures

- Performance measures: IR-system returns ranked list of answer documents
  → Precision and Recall ↝ Precision@rank , Recall@rank

Rank	Judgment	Precision_{Rank}	Recall_{Rank}
1	R	1.0	.11
2	N	.50	.11
3	R	.66	.22
4	N	.50	.22
5	R	.60	.33
6	R	.66	.44
7	N	.57	.44
8	R	.63	.55
9	N	.55	.55
10	N	.50	.55
11	R	.55	.66
12	N	.50	.66
13	N	.46	.66
14	N	.43	.66
15	R	.47	.77
16	N	.44	.77
17	N	.44	.77
18	R	.44	.88
19	N	.42	.88
20	N	.40	.88
21	N	.38	.88
22	N	.36	.88
23	N	.35	.88
24	N	.33	.88
25	R	.36	1.0

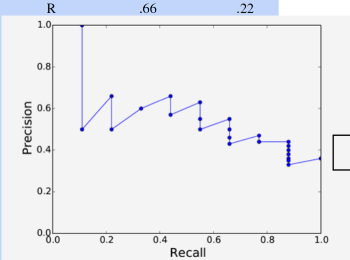
(assuming the collection has 9 relevant documents).



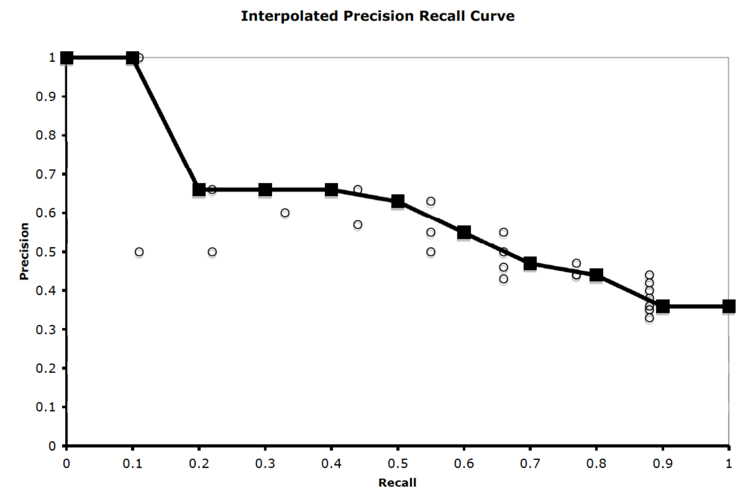
IR: Performance Measures

- Performance measures: IR-system returns ranked list of answer documents
  → Precision and Recall ↝ Precision@rank , Recall@rank

Rank	Judgment	$Precision_{Rank}$	$Recall_{Rank}$
1	R	1.0	.11
2	N	.50	.11
3	R	.66	.22
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18	R	.44	.88
19	N	.42	.88
20	N	.40	.88
21	N	.38	.88
22	N	.36	.88
23	N	.35	.88
24	N	.33	.88
25	R	.36	1.0



Interpolated Precision	Recall
1.0	0.0
1.0	.10
.66	.20
.66	.30
.66	.40
.63	.50
.55	.60
.47	.70
.44	.80
.36	.90
.36	1.0



IR: Performance Measures

- Performance measures: IR-system returns ranked list of answer documents
  → Precision and Recall ↝ Precision@rank , Recall@rank

Rank	Judgment	Precision_{Rank}	Recall_{Rank}
1	R	1.0	.11
2	N	.50	.11
3	R	.66	.22
4	N	.50	.22
5	R	.60	.33
6	R	.66	.44
7	N	.57	.44
8	R	.63	.55
9	N	.55	.55
10	N	.50	.55
11	R	.55	.66
12	N	.50	.66
13	N	.46	.66
14	N	.43	.66
15	R	.47	.77
16	N	.44	.77
17	N	.44	.77
18	R	.44	.88
19	N	.42	.88
20	N	.40	.88
21	N	.38	.88
22	N	.36	.88
23	N	.35	.88
24	N	.33	.88
25	R	.36	1.0

Mean average precision: in contrast to previous slide: only note precision where recall actually changes (in example: ranks 1,3,5,6,...)

$$\text{AP} = \frac{1}{|R_r|} \sum_{d \in R_r} \text{Precision}_r(d)$$

Set of revelant documents above rank r

$$\text{MAP} = \frac{1}{|Q|} \sum_{q \in Q} \text{AP}(q)$$

IR with Embeddings

- standard IR: query words and document words need to match (vocabulary mismatch problem)

- obvious idea:

$$h_q = \text{BERT}_Q(\text{q})\,[\text{CLS}]$$
$$h_d = \text{BERT}_D(\text{d})\,[\text{CLS}]$$
$$\text{score}(d,q) = h_q \cdot h_d$$

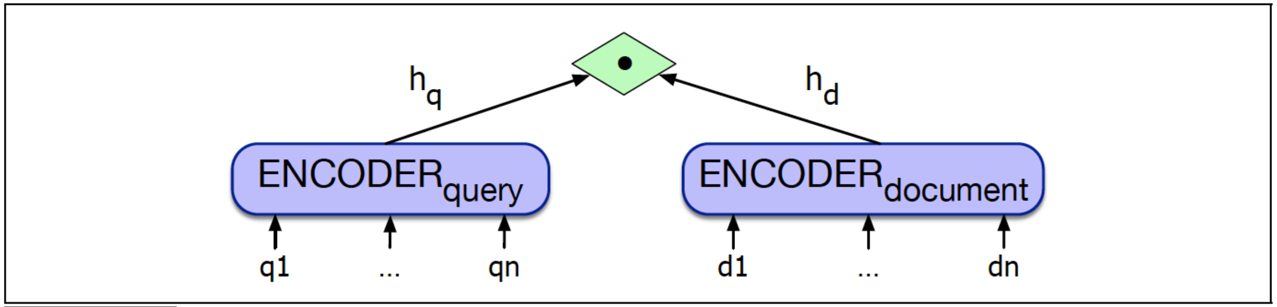


Figure 23.8 BERT bi-encoder for computing relevance of a document to a query.

obvious problem (Google scale)
compared to inverted index: performance

IR-based Factoid Question Answering

- Find short text fragments that are answers to question

Question	Answer
Where is the Louvre Museum located?	in Paris, France
What are the names of Odin's ravens?	Huginn and Muninn
What kind of nuts are used in marzipan?	almonds
What instrument did Max Roach play?	drums
What's the official language of Algeria?	Arabic

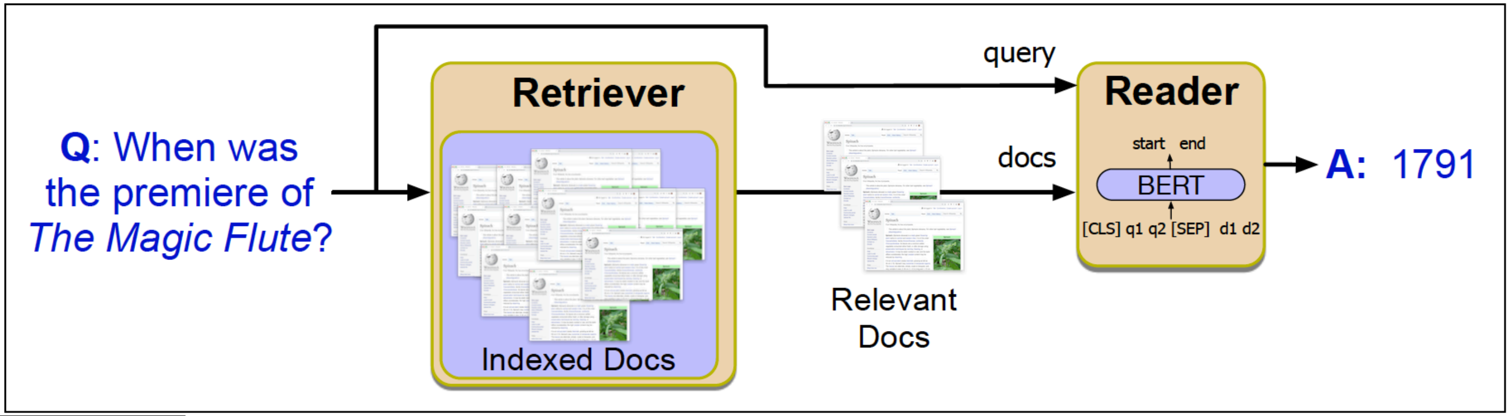


Figure 23.10 IR-based factoid question answering has two stages: **retrieval**, which returns relevant documents from the collection, and **reading**, in which a neural reading comprehension system extracts answer spans.

IR-based Factoid Question Answering: Datasets

- SQuAD 2.0 (2018): 150,000 questions from Wikipedia (human built)

Beyoncé Giselle Knowles-Carter (born September 4, 1981) is an American singer, songwriter, record producer and actress. Born and raised in Houston, Texas, she performed in various singing and dancing competitions as a child, and rose to fame in the late 1990s as lead singer of R&B girl-group Destiny's Child. Managed by her father, Mathew Knowles, the group became one of the world's best-selling girl groups of all time. Their hiatus saw the release of Beyoncé's debut album, Dangerously in Love (2003), which established her as a solo artist worldwide, earned five Grammy Awards and featured the Billboard Hot 100 number-one singles "Crazy in Love" and "Baby Boy".
Q: "In what city and state did Beyoncé grow up?" A: "Houston, Texas"
Q: "What areas did Beyoncé compete in when she was growing up?" A: "singing and dancing"
Q: "When did Beyoncé release Dangerously in Love?" A: "2003"

Figure 23.11 A (Wikipedia) passage from the SQuAD 2.0 dataset (Rajpurkar et al., 2018) with 3 sample questions and the labeled answer spans.

- Hotpot QA (2018): crowd worker made: derive questions that require a set of documents to answer

- Trivia QA (2017): 650,000 question-answer-evidence triples (evidence supporting web documents)

IR-based Factoid Question Answering: Datasets

- Natural Questions (2019): human annotator get actual Google query: destill long and short answer (incl. “no answer”) from Wikipedia page from top 5 IR results

- TyDi QA (2020): 204,000 q&a pairs from 11 typologically diverse languages

IR-Based QA: Reader (Answer Span Extraction)

- Span-labelling: q (n tokens), document passage p (m tokens): infer $p(a|p,q)$ for span a in p

start position (index) of span

end position (index) of span

Simplifying assumption: $p(a|p,q) = p_{start}(a_s|p,q)p_{end}(a_e|p,q)$

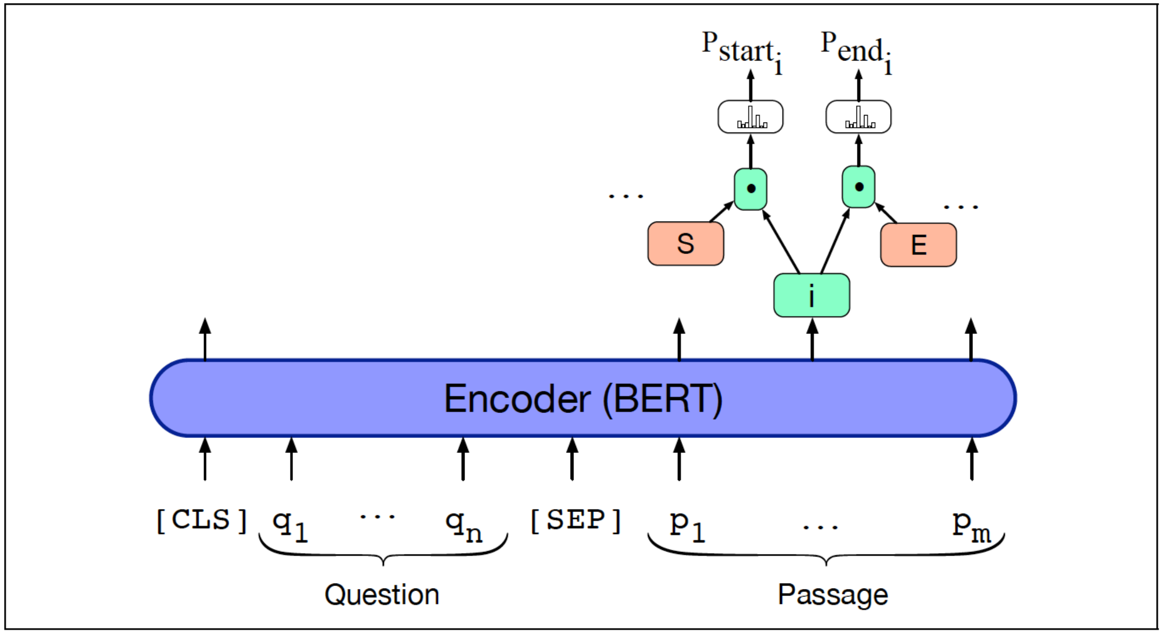


Figure 23.12 An encoder model (using BERT) for span-based question answering from reading-comprehension-based question answering tasks.

BERT embedding of i-th passage token

$$P_{start_i} = \frac{\exp(S \cdot p'_i)}{\sum_j \exp(S \cdot p'_j)}$$

$$P_{end_i} = \frac{\exp(E \cdot p'_i)}{\sum_j \exp(E \cdot p'_j)}$$

$$L = -\log P_{start_i}^{(Gold)} - \log P_{end_i}^{(Gold)}$$

The score of a candidate span from position i to j is $S \cdot p'_i + E \cdot p'_j$, and the highest scoring span in which $j \geq i$ is chosen is the model prediction.

Entity Linking

- Entity linking: associating a mention in text with some real-world entity in an ontology (e.g. Wikipedia (“wikification”), DBPedia, OWL-based…)

- components / stages: mention detection, mention disambiguation

Wikification example: TAGME (2011):

- index Wikipedia pages $\{e_i\}$ (page ↔ entity) in standard IR (e.g. Lucene)
- For each e_i compute in-link count
- extract anchor string from each in-link on other page

  Stanford University
- anchor dictionary entry for page/entity e_i : e_i’s title, all in-link anchor strings
- add to each anchor string: its overall ocurrence frequency $freq(a)$ (including non-anchor appearances) in all of Wikipedia, and overall occurrence frequency as anchor in links $link(a)$)
- delete anchor strings with low link probability $\frac{link(a)}{freq(a)}$

Entity Linking: TAGME (contd.)

- Mention detection: for each question token sequence of up to 6 tokens: query anchor dictionary → finds mention spans in anchor strings

- if mention span is in anchor string pointing to only one Wikipedia page: done! If not: Mention disambiguation:

  for all pages $\mathcal{E}(a)$ pointed to by ambiguous anchor span a

  - prior probability that anchor string a points to entity (page) $e \in \mathcal{E}(a)$

    $$\text{prior}(a \to e) \;=\; p(e|a) = \frac{\text{count}(a \to e)}{\text{link}(a)}$$

    example question: *What Chinese Dynasty came before the Yuan?*

    problem: p(entity:yuan_currency | *Yuan*) > p(entity:yuan_dynasty | *Yuan*)

  - → relatedness score: incorporate other candidate anchor spans:

    example: *Chinese Dynasty* → page:Dynasties_in_Chinese_history

    for each candidate anchor span a in q, compute relatedness score to all entities $e \in \mathcal{E}(a)$: relatedness score of the link $a \to e$ is the weighted average relatedness between e and all other (pointed to) entities e' in q.

Entity Linking: TAGME (contd.)

for each candidate anchor span a in q, compute relatedness score to all entities $e \in \mathcal{E}(a)$: relatedness score of the link $a \rightarrow e$ is the weighted average relatedness between e and all other (pointed to) entities e' in q.

i.e. for A, B: entities and $in(A)$: set of entities (pages) pointing to A:

$$\mathrm{rel}(A,B) = \frac{\log(\max(|\mathrm{in}(A)|, |\mathrm{in}(B)|)) - \log(|\mathrm{in}(A) \cap \mathrm{in}(B)|)}{\log(|W|) - \log(\min(|\mathrm{in}(A)|, |\mathrm{in}(B)|))}$$

the vote given by anchor b to the candidate annotation $a \rightarrow X$

$$\mathrm{vote}(b,X) = \frac{1}{|\mathcal{E}(b)|} \sum_{Y \in \mathcal{E}(b)} \mathrm{rel}(X,Y) p(Y|b)$$

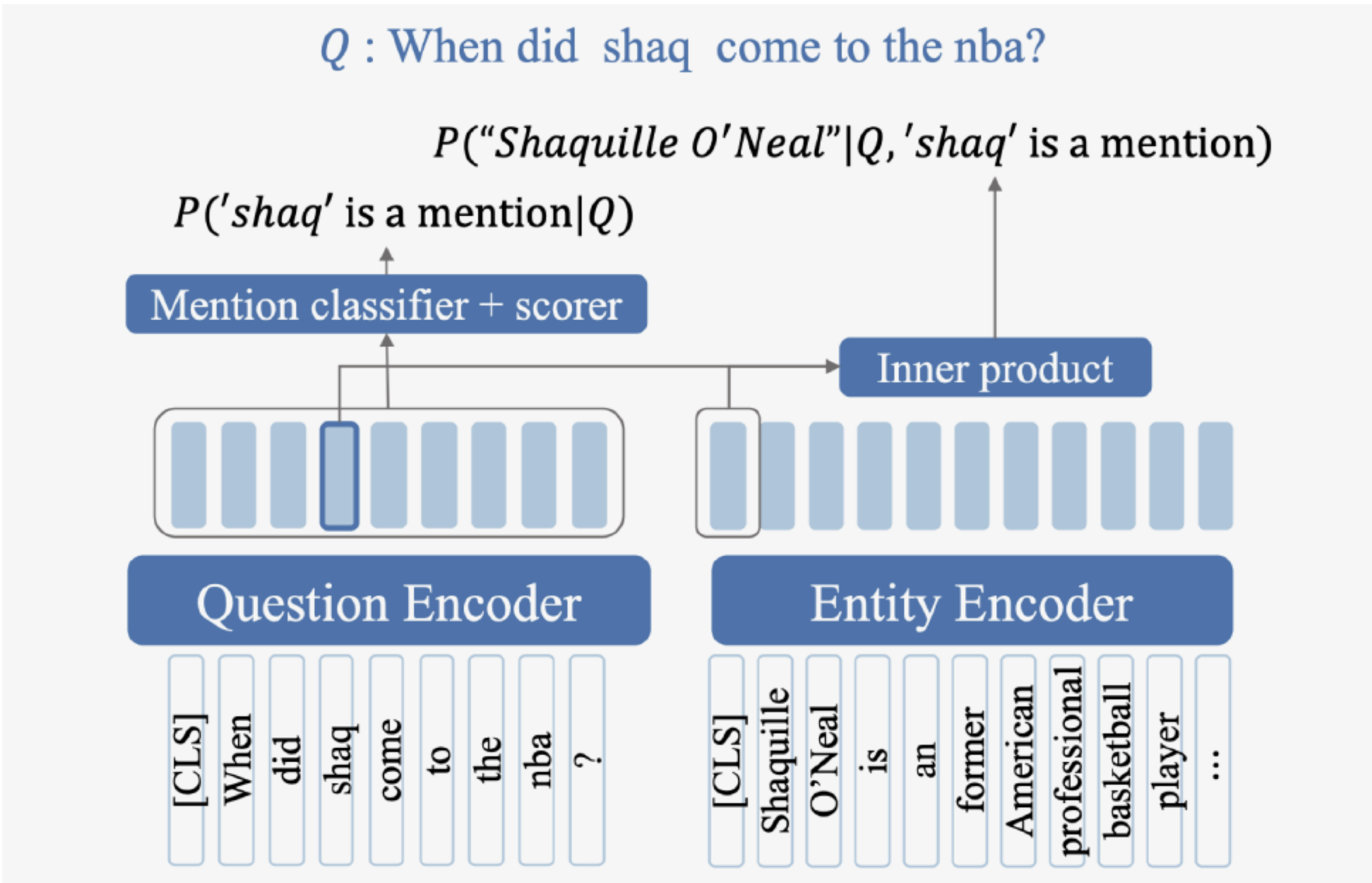
→ relatedness score for $a \rightarrow X$

$$\mathrm{relatedness}(a \rightarrow X) = \sum_{b \in \mathcal{X}_q \setminus a} \mathrm{vote}(b,X)$$

- for question q choose answer entity with highest prior probability among entities with highest relatedness scores

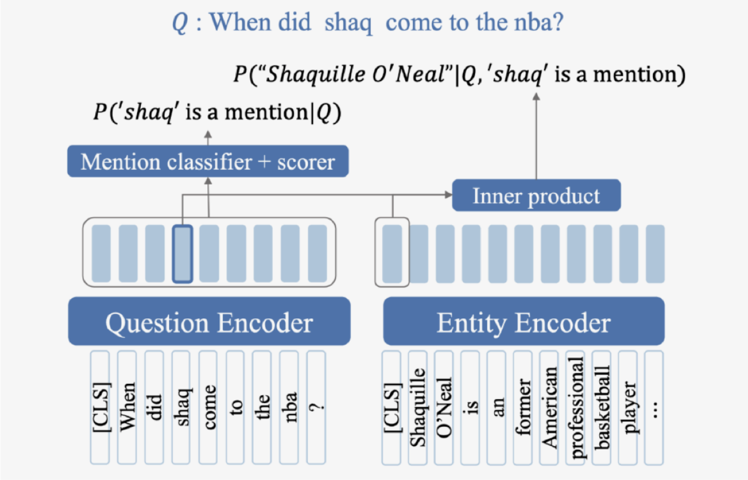
Neural Graph-Based Linking

ELQ (2020):



Neural Graph-Based Linking

ELQ (2020): Entity Mention Detection



$$[\mathbf{q}_1 \cdots \mathbf{q}_n] = \mathrm{BERT}([\mathrm{CLS}] q_1 \cdots q_n [\mathrm{SEP}])$$

$$s_{\mathrm{start}}(i) = \mathbf{w}_{\mathrm{start}} \cdot \mathbf{q}_i$$

$$s_{\mathrm{mention}}(t) = \mathbf{w}_{\mathrm{mention}} \cdot \mathbf{q}_t$$

$$s_{\mathrm{end}}(j) = \mathbf{w}_{\mathrm{end}} \cdot \mathbf{q}_j$$

$$p([i,j]) = \sigma\left(s_{\mathrm{start}}(i) + s_{\mathrm{end}}(j) + \sum_{t=i}^{j} s_{\mathrm{mention}}(t)\right)$$

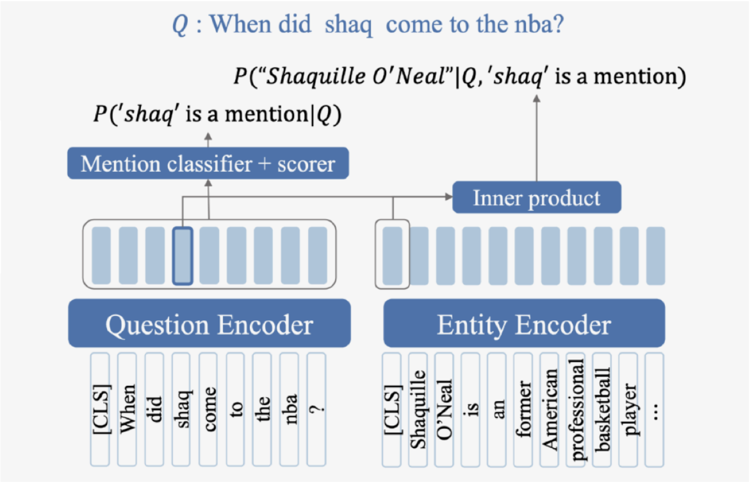
Mention detection loss:

$$\mathcal{L}_{\mathrm{MD}} = -\frac{1}{N} \sum_{1 \le i \le j \le min(i+L-1,n)} \left(y_{[i,j]} \log p([i,j]) + (1 - y_{[i,j]}) \log(1 - p([i,j]))\right)$$

= 1 if $[i, j]$ is gold mention

Neural Graph-Based Linking

ELQ (2020): Entity Linking



entity encoding:

$$\mathbf{x}_e = \mathrm{BERT}_{[\mathrm{CLS}]}([\mathrm{CLS}]t(e_i)[\mathrm{ENT}]d(e_i)[\mathrm{SEP}])$$

title of entity (page)

first 128 tokens of entity (page))

span $[i, j]$ encoding:

$$\mathbf{y}_{i,j} = \frac{1}{(j-i+1)} \sum_{t-i}^{j} \mathbf{q}_t$$

linking:

$$s(e, [i, j]) = \mathbf{x}_e^{\cdot}\mathbf{y}_{i,j} \qquad p(e|[i, j]) = \frac{\exp(s(e, [i, j]))}{\sum_{e' \in \mathcal{E}} \exp(s(e', [i, j]))}$$

Entity linking loss:

$$\mathcal{L}_{\mathrm{ED}} = -logp(e_g|[i, j])$$

gold entity

Knowledge-Based QA from RDF Triple Stores

- RDF triple:

subject	predicate	object
Ada Lovelace	birth-year	1815

- possible questions: *When was Ada Lovelace born?*

  Who was born in 1815?

- Entity linking: as before

- Relation detection, linking and disambiguation:

"When was Ada Lovelace born?" $\rightarrow$ `birth-year (Ada Lovelace, ?x)`

"What is the capital of England?" $\rightarrow$ `capital-city(?x, England)`

$$\mathbf{m}_r = \mathrm{BERT}_{\mathrm{CLS}}([\mathrm{CLS}]q_1 \cdots q_n[\mathrm{SEP}])$$

$$s(\mathbf{m_r}, r_i) = \mathbf{m_r} \cdot \mathbf{w_{r_i}}$$

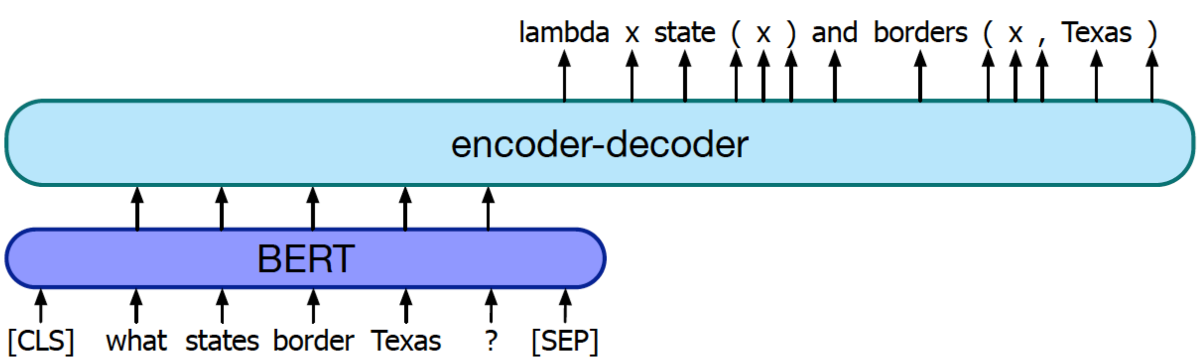
$$p(r_i|q_1, \cdots, q_n) = \frac{\exp(s(\mathbf{m_r}, r_i))}{\sum_{k=1} N_R \exp(s(\mathbf{m_r}, r_k))}$$

Trained representation for each relatipn r_i out of a fixed set of relations (e.g. rdf predicates)

Knowledge-Based QA by Semantic Parsing

- Map question to logical form (predicate logic, SQL, SPARQL etc.) (supervised)

Question	Logical form
What states border Texas?	$\lambda x.\text{state}(x) \wedge \text{borders}(x, \text{texas})$
What is the largest state?	$\text{argmax}(\lambda x.\text{state}(x), \lambda x.\text{size}(x))$
I'd like to book a flight from San Diego to Toronto	SELECT DISTINCT f1.flight_id FROM flight f1, airport_service a1, city c1, airport_service a2, city c2 WHERE f1.from_airport=a1.airport_code AND a1.city_code=c1.city_code AND c1.city_name= 'san diego' AND f1.to_airport=a2.airport_code AND a2.city_code=c2.city_code AND c2.city_name= 'toronto'
How many people survived the sinking of the Titanic?	`(count (!fb:event.disaster.survivors fb:en.sinking_of_the_titanic))`
How many yards longer was Johnson's longest touchdown compared to his shortest touchdown of the first quarter?	ARITHMETIC diff(SELECT num(ARGMAX(SELECT)) SELECT num(ARGMIN(FILTER(SELECT))))



Knowledge-Based QA with Language Models

- “ask the language model (T5, Chat-GPT etc) directly”:

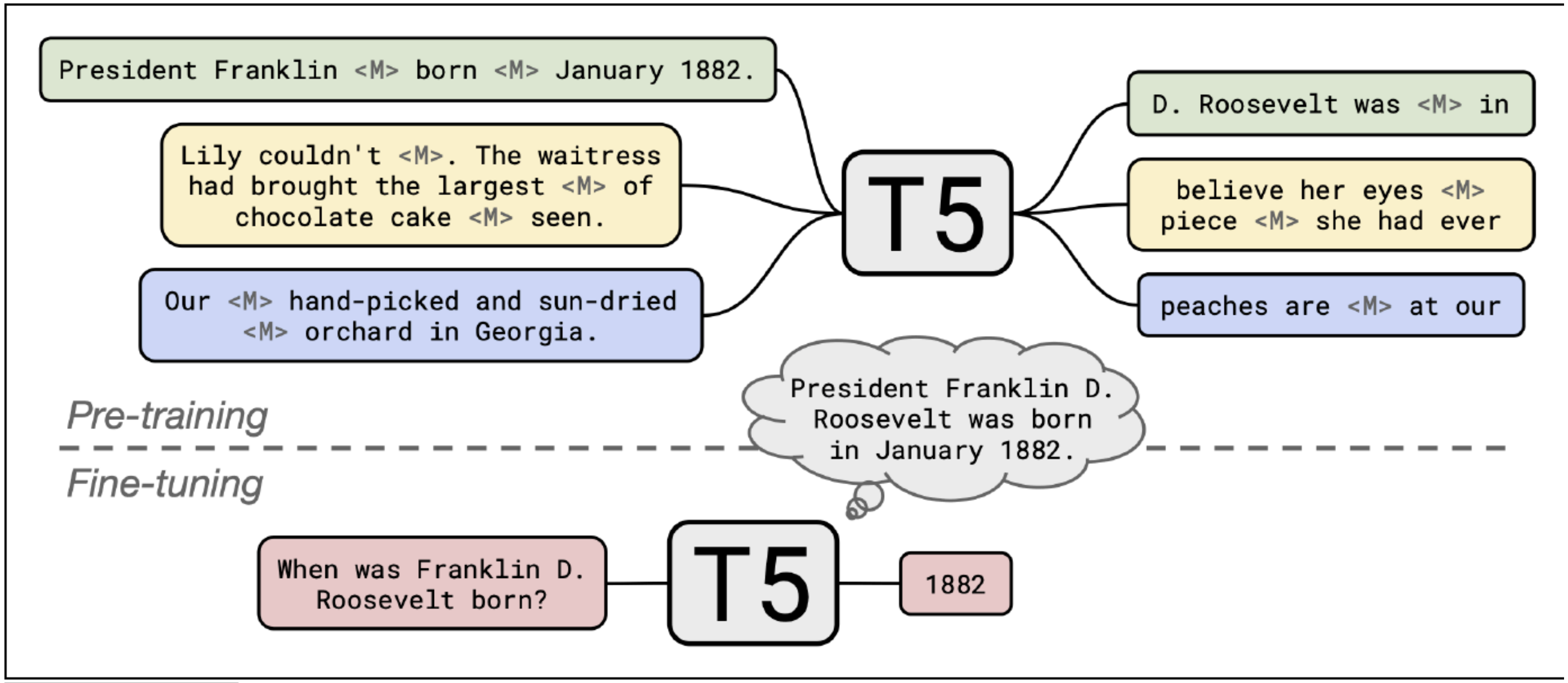


Figure 23.16 The T5 system is an encoder-decoder architecture. In pretraining, it learns to fill in masked spans of task (marked by <M>) by generating the missing spans (separated by <M>) in the decoder. It is then fine-tuned on QA datasets, given the question, without adding any additional context or passages. Figure from Roberts et al. (2020).

Bibliography

(1) Dan Jurafsky and James Martin: Speech and Language Processing (3rd ed. draft, version Jan 2022); Online: https://web.stanford.edu/~jurafsky/slp3/ (URL, Oct 2022); this slide-set is especially based on chapter 23

Recommendations for Studying

- minimal approach:

  work with the slides and understand their contents! Think beyond instead of merely memorizing the contents

- standard approach:

  minimal approach + read the corresponding pages in Jurafsky [1]

- interested students

  == standard approach