QUESTION ANSWERING

Presented by

Robin Hilbrecht, David Novak,
Simon Pycha, Anna Schmer

Content

Preamble

Information retrieval (IR)

- Term weighting
- Document scoring
- Inverted Index
- Evaluation of IR-Systems
- IR with dense vectors

IR-based factoid question answering

- IR-based QA Dataset
- IR-phases

Entity Linking

PREAMBLE

„The quest for knowledge is deeply human, […]“

- Two major paradigms were used by the early 1960s: **information-retrieval-based** and **knowledge-based**
- Question Answering (QA) Systems are used when interacting with a virtual assistant or a search engine or querying a database to fill human information needs
- Most QA Systems focus on **factoid questions** (E.g. *When does the Semester start?*, *How much does an alpaca cost?*)
- IR-based QA (open domain QA) relies on the vast amount of text on the web or in collections of scientific papers
  - Used to find relevant passages based on a given question
  - Neural reading comprehension algorithms read retrieved passages draw answers from spans of text
  - Key component (relevant throughout NLP): IR
- Knowledge-based QA uses a system build semantic representation of the query to map questions to their respective logical representation, representations are then used to query databases
  - E.g. *When was Ada Lovelace born?* to the gapped relation *birth-year (Ada Lovelace ?x)*
  - Key component (relevant throughout NLP): Entity Linking

INFORMATION RETRIEVAL (IR)

- Retrieval of all manner of media based on user information need
- Resulting IR system is often called search engine
- User poses query to retrieval system, which then returns an ordered set of documents
- Basic IR architecture uses vector space model (in this case also bag-of-words model)

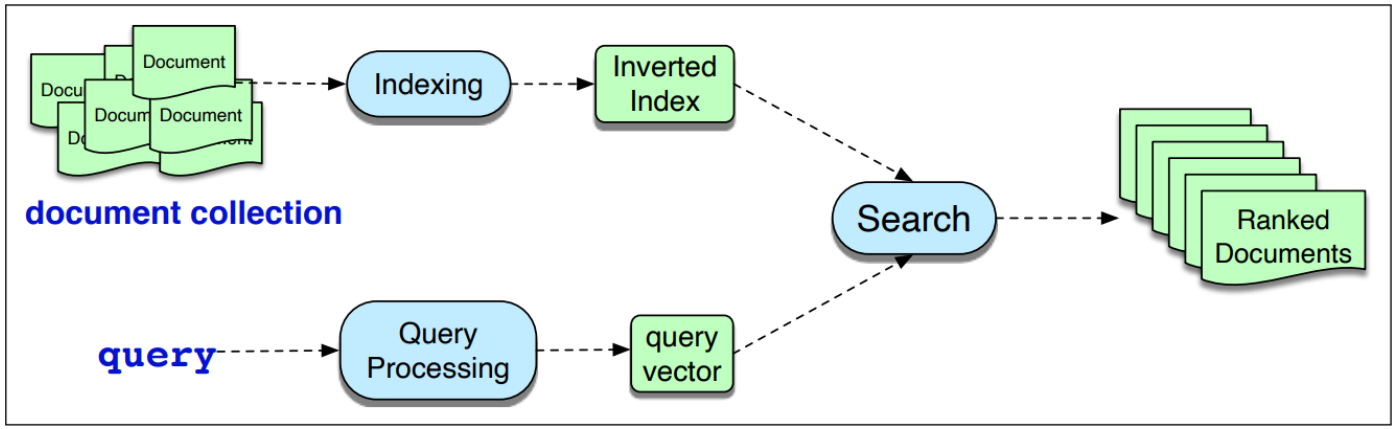


Fig. 1 The architecture of an ad hoc IR system

IR – Term weighting

tf-idf as product of the term frequency (tf) and the inverse document frequency (idf)

Term frequency

- words that appear more often are likely to be informative

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

Inverse document frequency

- Terms that occur only in few documents are useful to discriminating those documents

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

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

IR - Document scoring

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

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

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

Example Query and Documents:

Query: sweet love

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

Computation of tf-idf for Doc 1 and Doc 2:

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

Ranking Documents:

Doc	$\lvert d \rvert$	**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

IR – Document scoring

- The slightly more complex BM25 (Okapi BM25 after Okapi IR in which it was introduced) weighting scheme adds the parameter **k**, to adjust the balance between tf and idf, and **b**, which controls the importance of document length normalization

$$\text{BM25 score} = \sum_{t \in q} \overbrace{\log\left(\frac{N}{df_t}\right)}^{\text{IDF}} \overbrace{\frac{tf_{t,d}}{k\left(1-b+b\left(\frac{|d|}{|d_{\text{avg}}|}\right)\right)+tf_{t,d}}}^{\text{weighted tf}}$$

Stop words: While in the past it was common to remove high-frequency words from the query and document before representing them, modern IR systems are able to downweight function words (idf weighting)

IR - Inverted Index

how {1} → 3 [1]
is {1} → 3 [1]
love {2} → 1 [1] → 3 [1]
nurse {2} → 1 [1] → 4 [1]
sorry {1} → 2 [1]
sweet {3} → 1 [2] → 2 [1] → 3 [1]

- efficiently find documents that contain words in the query
- gives a list of documents that contain the postings term
  - It consists of two parts
- Dictionary
  - list of terms, each pointing to a postings list for the term
- Postings
  - list is the list of document IDs associated with each term
  - can contain information like the term frequency or the exact positions of terms in the document

IR - Evaluation of IR-Systems

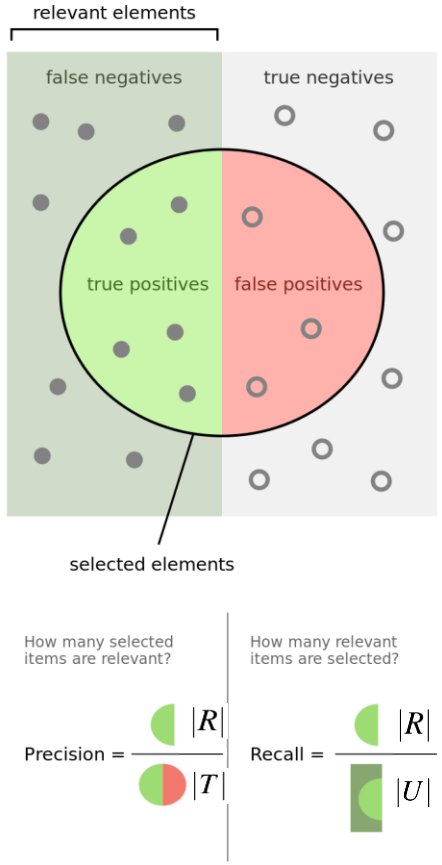


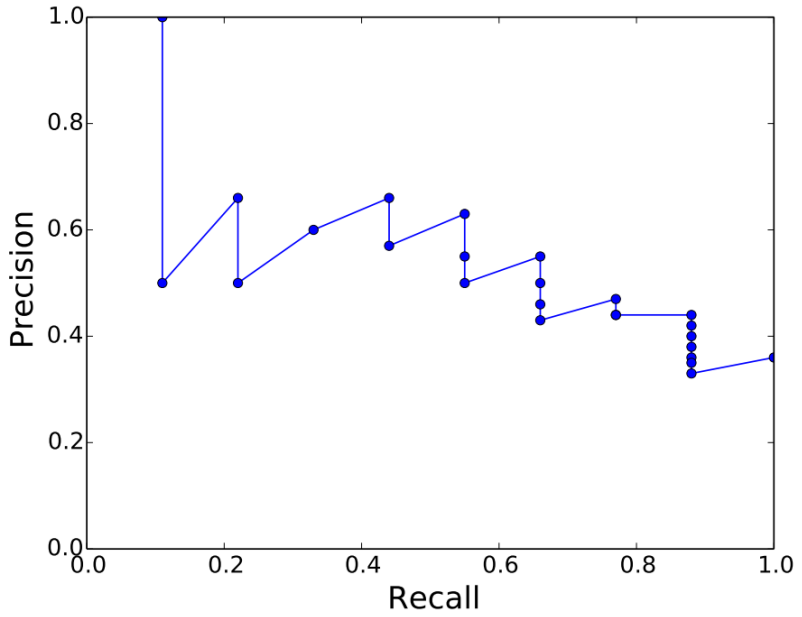
Fig 2 Precision and Recall Scheme (Quelle: wikipedia)

$$Precision = \frac{|R|}{|T|} \qquad Recall = \frac{|R|}{|U|}$$

- To measure the performance of ranked retrieval systems we use precision and recall metrics
- Precision: the fraction of the returned documents that are relevant
- Recall: the fraction of all relevant documents that are returned

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

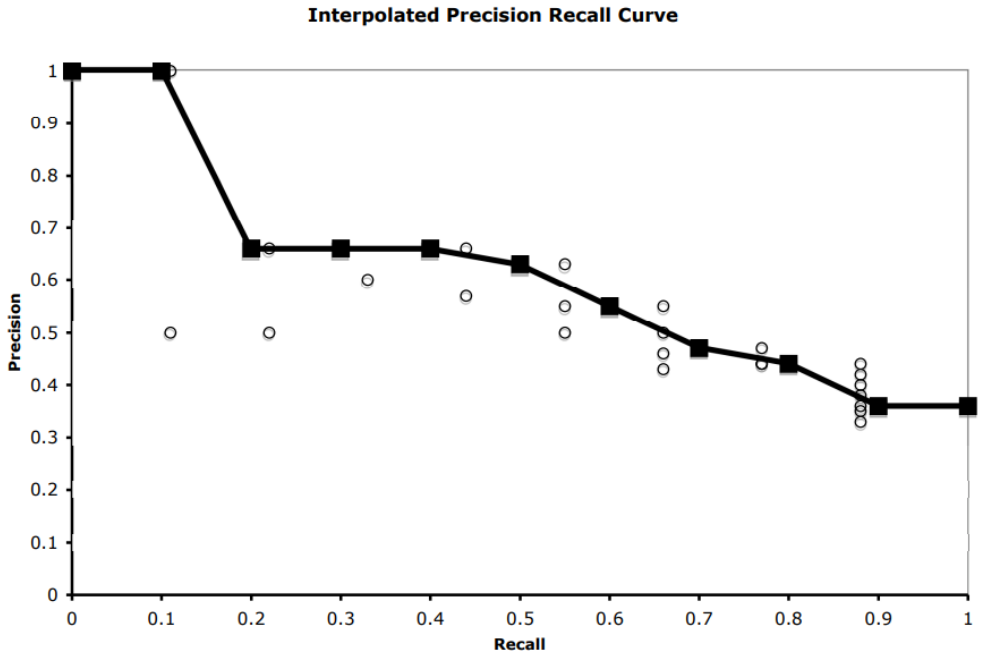
(assuming the collection has 9 relevant documents)



IR – interpolated precision und mAP

$$\text{IntPrecision}(r) = \max_{i>=r} \text{Precision}(i)$$

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 with dense vectors

Problem: tf-idf und BM25 für IR nur funktionieren, wenn Wörter sich in den Abfragen und Dokumenten überschneiden.
Die exakten Worte müssen in query formuliert werden.
Lösung: Synonymie umgehen!

→ Vokabulary mismatch problem

Dichte Einbettungen bei Moderne Methode
"BERT Encoders" (biencoder) verwendet
Ziel: Sprachmodell generieren
2 getrennte Encoder-Modelle
1. Kodierung der Anfrage
2. Kodierung des Dokuments
3. = Skalarprodukt wird als **SCORE** zwischen zwei Vektoren benutzt

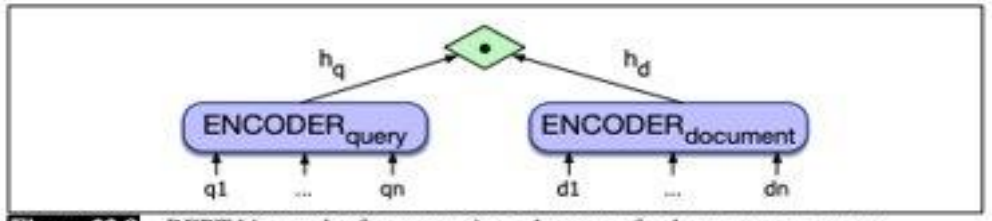


Figure 23.8 BERT bi-encoder for computing relevance of a document to a query.

Anfrage und Dokument als [CLS-Token] darstellen

Komplexere Version für Darstellung von kodiertem Text:
Ø-Pooling über BERT Ausgänge aller Tokens, anstelle des CLS-Tokens;
Zusätzliche Gewichtsmatrizen nach Encoding oder Punkt Produkten hinzufügen

Dense Vectors und **QA** sind noch Forschungsgebiete
- Abstimmung der Encoder-Module für IR-Anfragen (Negativbeispiele erhalten mit Query-document combinations)
- Umgehen können, dass Dok. länger sind als Encoder (BERT) überhaupt verarbeiten können
- **Effizienz** – Jedes Dok. nach Ähnlichkeit der Abfrage bewerten

IR-BASED FACTOID QUESTION ANSWERING

IR-based factoid question answering (Open Domain QA)

Ziel: Frage des Nutzers zu beantworten, indem Textabschnitte aus dem Web oder großen Datenmengen gefunden werden!

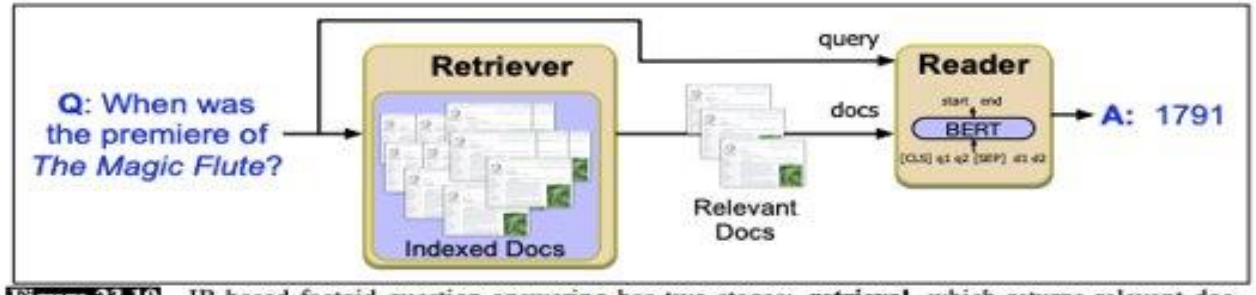


Figure 23.10 IR-based factoid question answering has two stages: **retrieval**, which returns relevant documents from the collection, and **reading**, in which a neural reading comprehension system extracts answer spans.

Dominates 2-Stufen-Modell " retrieve and read"

1. Relevante Passagen aus Kollektion werden abgerufen (oft mit Hilfe einer Suchmaschiene)
2. Ein **" reading comprehension algorithm"** durchläuft jede Passage und findet Abschnitte, die die Frage beantworten können

Beispiel:

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 23.9 Some factoid questions and their answers.

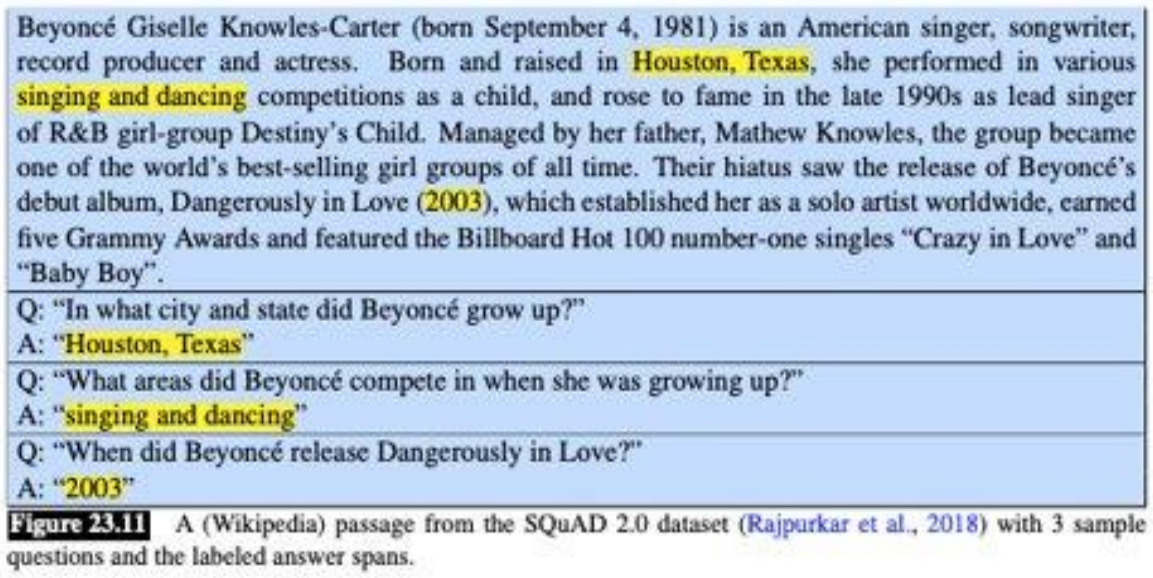
" reading comprehension systems" bekommen factoid question **q** und eine Passage **p**

- Kann Antwort enthalten und s zurück geben
- Keine Antwort
- Reihe von möglichen Antworten
- Antwort entspricht nicht dem Informationsbedarf des Nutzers

IR-based QA Datasets

- Erstes Leseverständnis – Tupel aus Textstelle, Frage und Antwort
- **"reading comprehension systems"** können Datensätze verwenden, um Reader zu trainieren (erhält Frage und Passage und gibt eine Antwort aus einer Passage)
- **"reading comprehension systems"** brauchen kein IR, da die Passage bereits einbezogen wurde

IR-based QA Datasets



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.

- SQUAD "Standard Question Answering Dataset"
- SQUAD 2.0
- HotpotQA
- TriviaQA
- TyDiQA

Morphologische Unterschiede, Wortsegmentierungen und mehrere Alphabeten stellen QA-Systeme vor Herausforderungen!

IR-phases

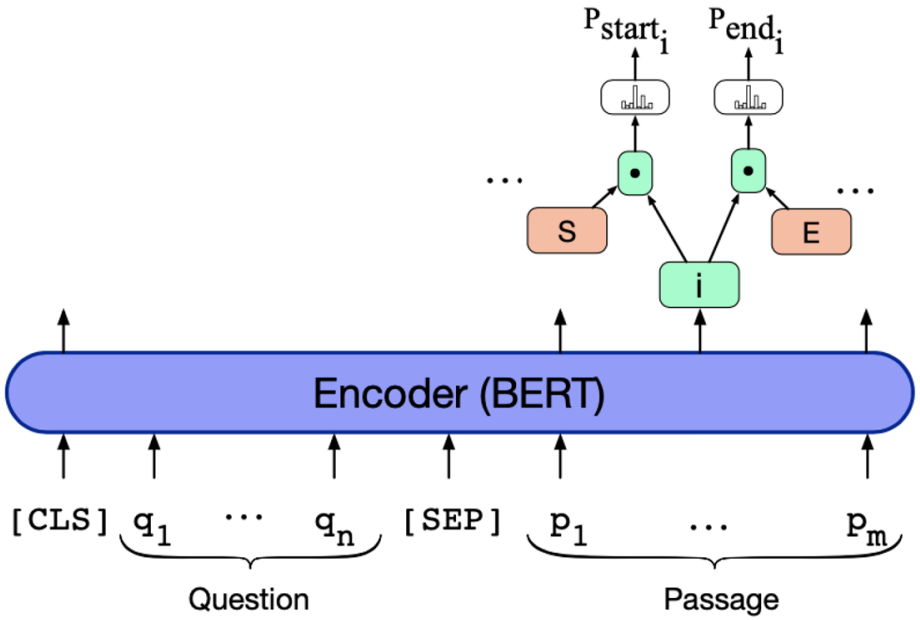
Phasen der IR-basierten Fragenbeantwortung:

1. **Retriever** = Bearbeitet entsprechende Informationen und macht diese verfügbar

2. **Reader** = Identifiziert und berechnet durch **span labeling** Wert möglicher Antworten

Verschiedene Spannweiten-Wahrscheinlichkeiten werden kalkuliert und von BERT entkodiert

-> Die Spanne mit der höchsten Wahrscheinlichkeit wird ausgespielt



ENTITY LINKING

Entity Linking

(Entitätsverknüpfungen)

Verbindung von Wortgebrauch und Korrelation mit entsprechender weltlicher Repräsentation

Wikification: Jede Seite der Wikipedia repräsentiert eine Entität, die mit der User-Anfrage übereinstimmen muss

Phasen der Entitätsverknüpfung:

1. Mention detection: Mithilfe eines angelegten Wörterbuchs, das die Verlinkungen zwischen den Wikipedia-Seiten als Such-Fundament benutzt werden Such-Querys gekürzt und gewichtet.

(„Wann wurde Amadeus Mozart geboren?“ -> Gewichtung für Mozart, mögliche Gewichtung für Amadeus)

2. Mention disambiguation: Ist eine Entität einzigartig, müssen keine weiteren Operationen vorgenommen werden

Liegt eine Ambiguität vor, erfolgt ein Ranking nach Prioritätswahrscheinlichkeit und Verwandtschaft/Zusammenhang

Verwandtschaft: Einbezug der Verbindung von Termen in der Suchanfrage und gemeinsamer Gewichtung

Yuan -> Währung, Nachname, Sprache

q: Welche Währung gab es vor dem Yuan? -> Verbindung von Yuan und Währung

-> Gewichtung bei der entsprechend Wiki-Seite, die höchste Gewichtung gewinnt

THANK YOU FOR YOUR ATTENTION!

Sources

- Jurafsky, Dan; Martin, James H.: Question Answering. In: Speech and Language Processing (3rd ed. draft). (2021), pp. 470-486
- https://en.wikipedia.org/wiki/Precision_and_recall#/media/File:Precisionrecall.svg(Stand 29.10.2021)