

# Verbalizations of Entity-based Answers for Question Answering Systems

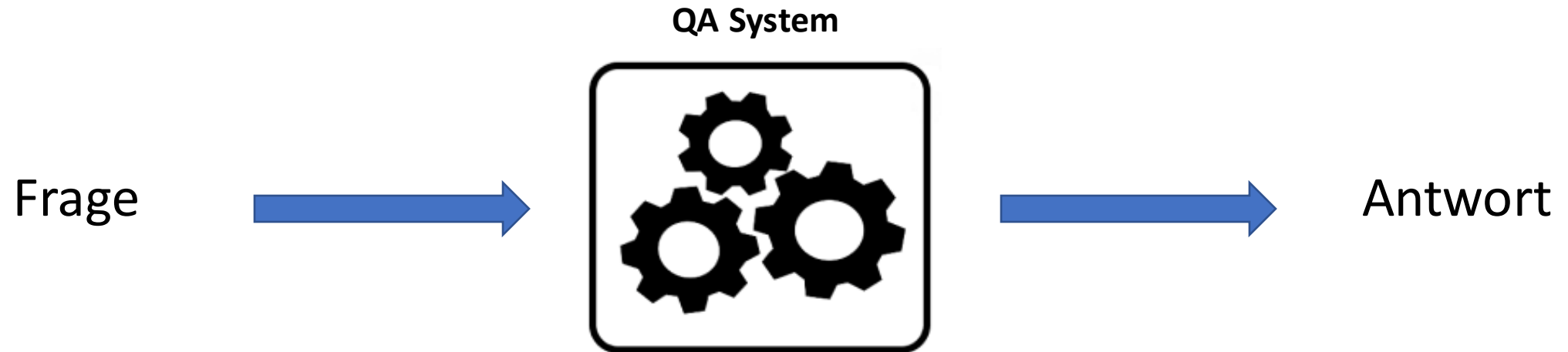
---

Deniz Simsek

Supervisor: Marcel Gohsen & Johannes Kiesel

## Verbalizations of Entity-based Answers for Question Answering Systems

# Question Answering System



## Verbalizations of **Entity-based Answers** for Question Answering Systems

# Entitäten

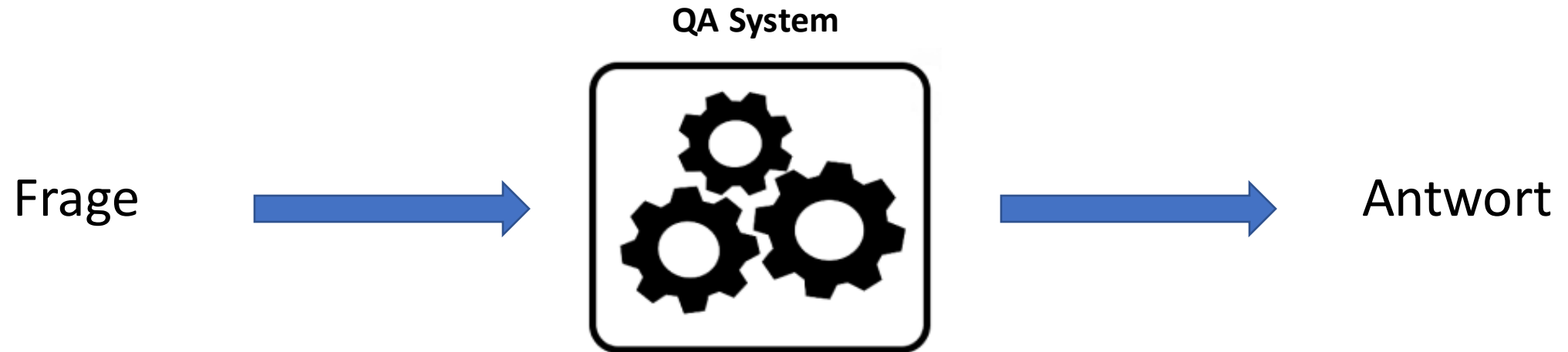
- Was ist eine Entität?
  - "Etwas, das unabhängig von anderen Dingen existiert und seine eigene unabhängige Existenz hat."<sup>1</sup>
  - Wichtig für Bedeutung + relevanten Informationen von Texten
  - Gehören zu Kategorien wie: Namen, Orte, Konzepte...

**"How tall is the Eiffeltower?"**

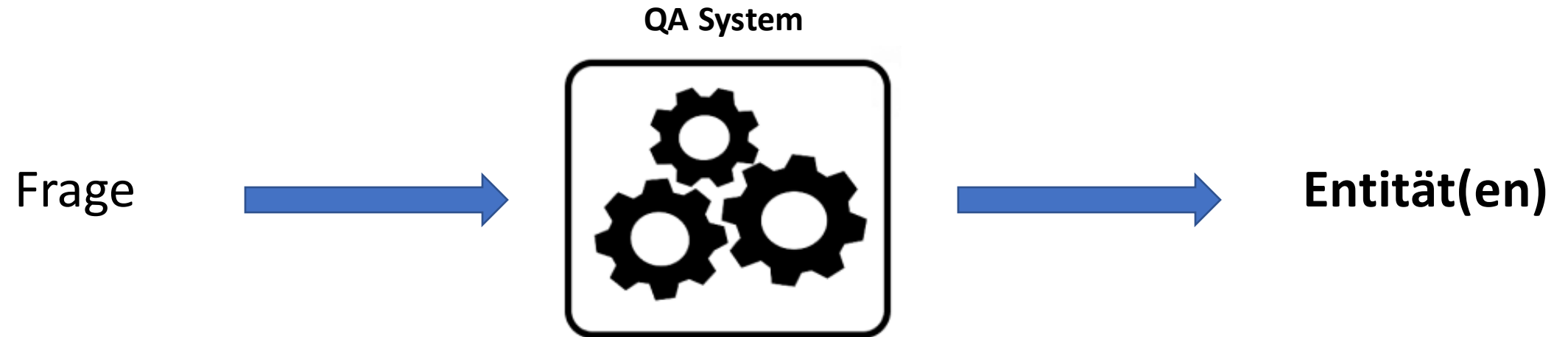
**"How tall is **the Eiffeltower**?"**

<sup>1</sup>. <https://dictionary.cambridge.org/dictionary/english/entity>

# Warum Entitäten?



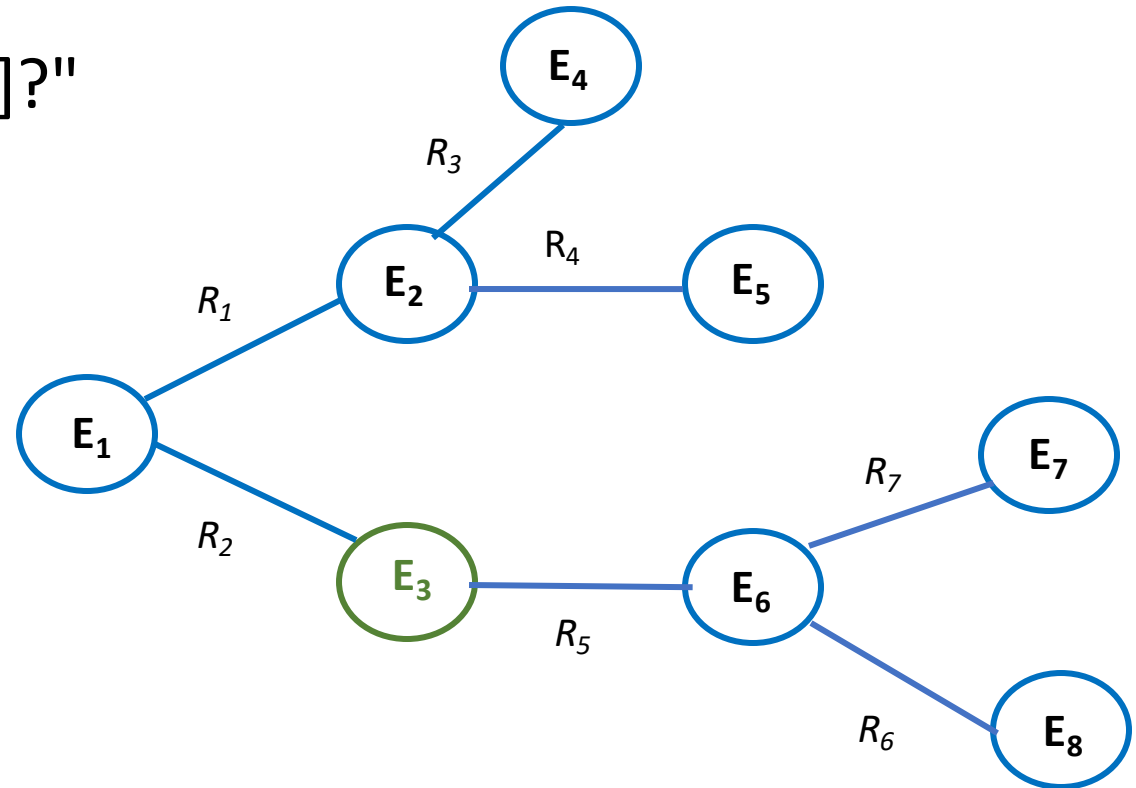
# Entitätsbasierte Antworten



# Einfaches Beispiel Knowledge Base Question Answering System

Frage: "What country borders [E<sub>1</sub>]?"

*Schritt 1: Topic-Entität finden*

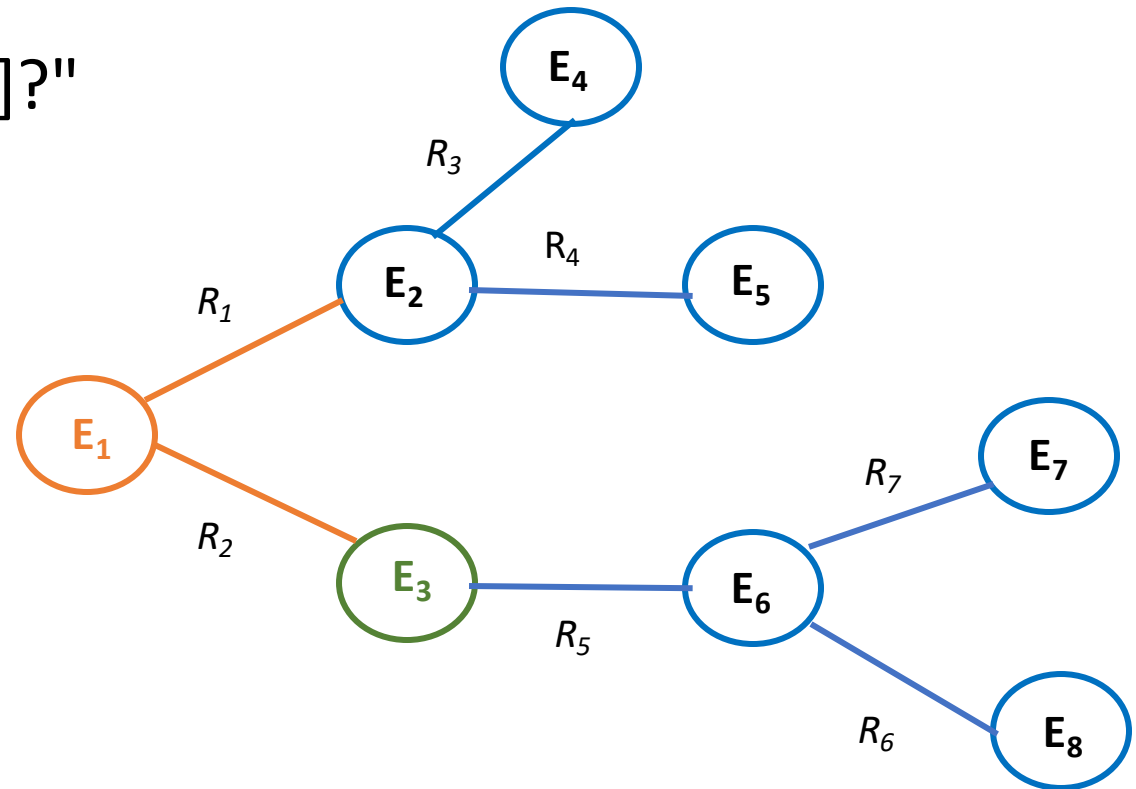




# Einfaches Beispiel Knowledge Base Question Answering System

Frage: "What country borders [E<sub>1</sub>]?"

*Schritt 2: Antwort Entität finden*



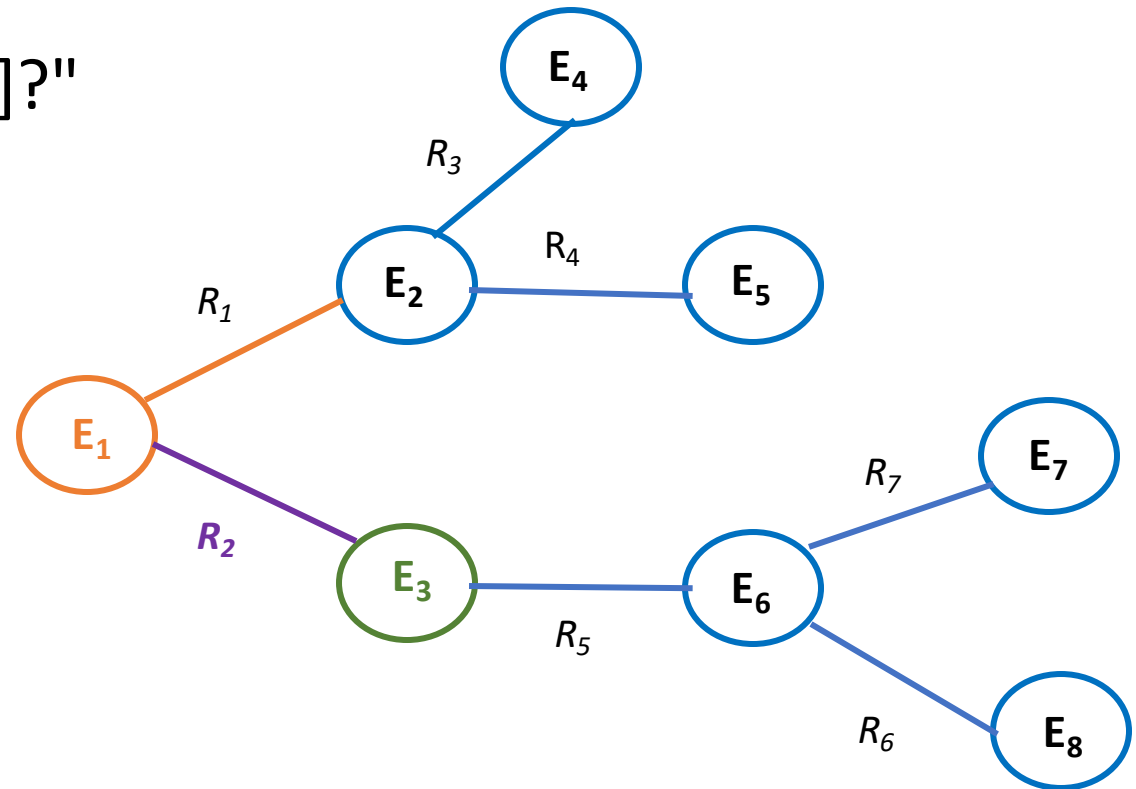
# Simple Beispiel Knowledge Base Question Answering System

Frage: "What country borders [E<sub>1</sub>]?"

Schritt 2.1: Semantisches Parsen

(E<sub>1</sub>, R<sub>2</sub>, E<sub>3</sub>)

Antwort: E<sub>3</sub>



# Gibt es nicht schon ChatGPT?

- Was ist mit ChatGPT?



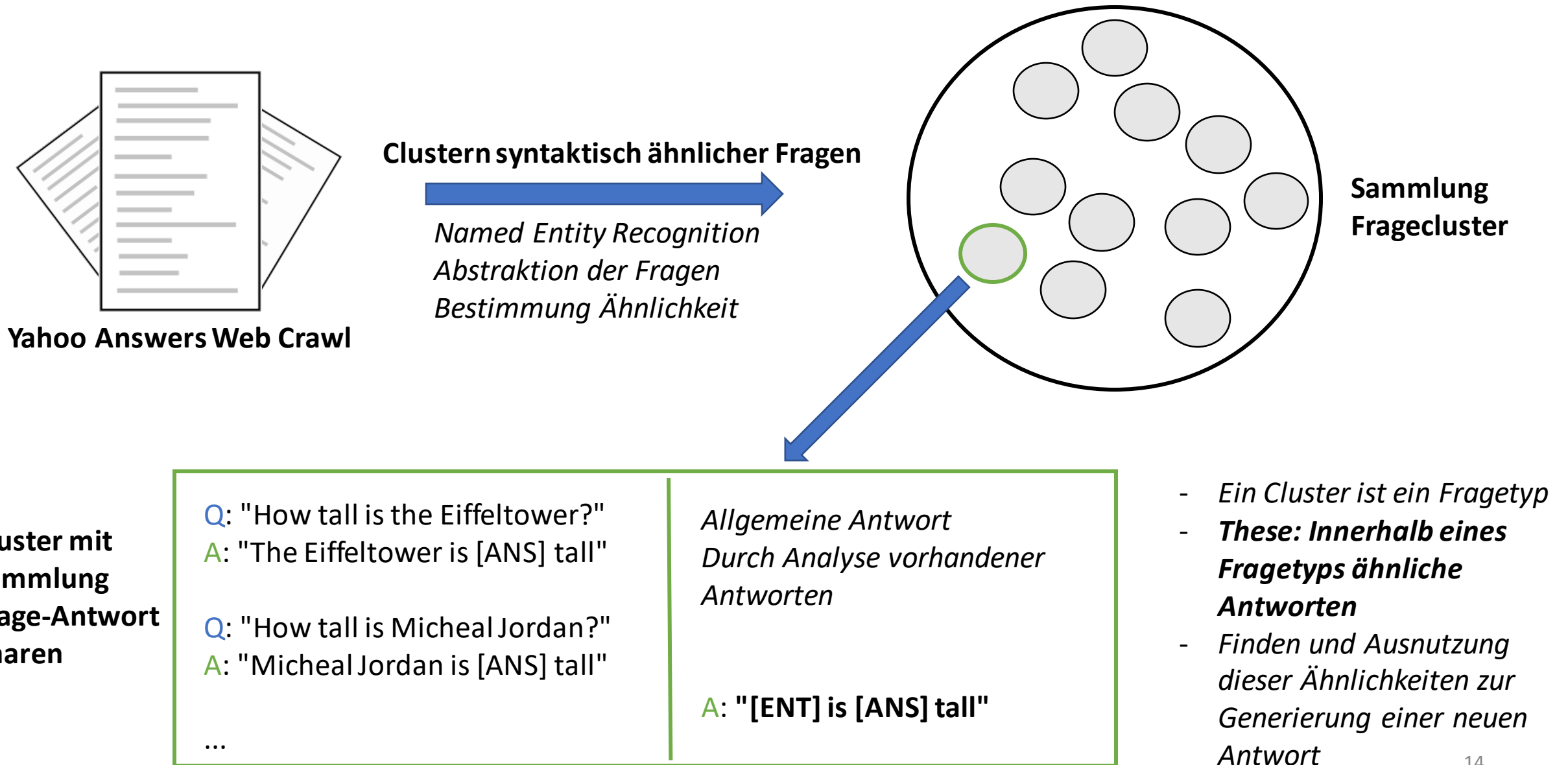
- Mein Ansatz
  - Grundgedanke: Verwendung Ausgangsfrage und Knowledge Graph (Entitäten) für Verbalisierung
  - Meine These: Ähnliche Fragen werden von ähnlichen Antworten beantwortet

# Verwandte Arbeiten

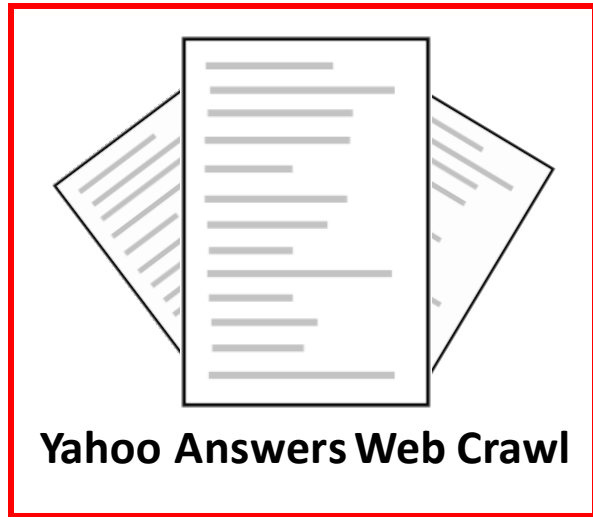
- Tatsächlich noch nicht allzu viel Forschung zu Verbalisierung solcher Antworten
- Bisherige Forschungsansätze
  - *VOGUE: Answer Verbalization through Multi-Task Learning (Kacupaj et al. ,2021)*

## Verbalizations of Entity-based Answers for Question Answering Systems

# Der 1. Ansatz (vereinfacht)



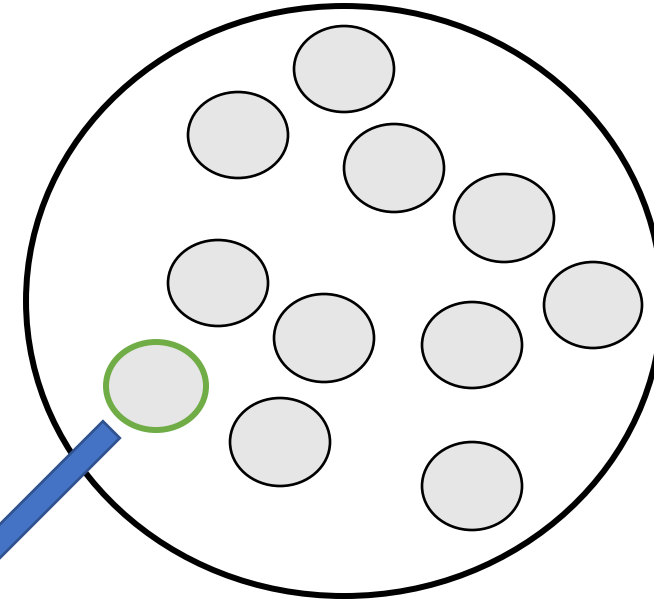
# Der 1. Ansatz (vereinfacht)



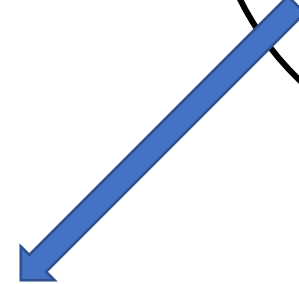
Clustern syntaktisch ähnlicher Fragen



*Named Entity Recognition*  
*Abstraktion der Fragen*  
*Bestimmung Ähnlichkeit*



Sammlung  
Fragecluster



Cluster mit  
Sammlung  
Frage-Antwort  
Paaren

Q: "How tall is the Eiffeltower?"

A: "The Eiffeltower is [ANS] tall"

Q: "How tall is Micheal Jordan?"

A: "Micheal Jordan is [ANS] tall"

...

*Allgemeine Antwort*  
*Durch Analyse vorhandener*  
*Antworten*

A: "[ENT] is [ANS] tall"

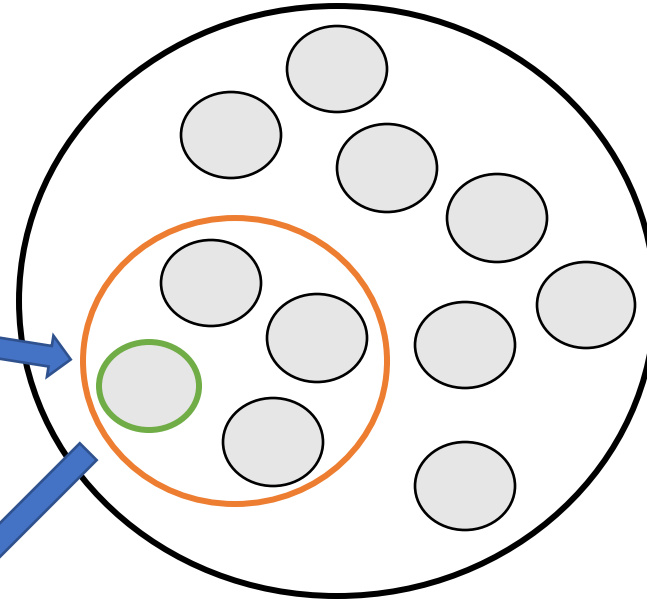
- Ein Cluster ist ein Fragetyp
- These: Innerhalb eines Fragetyps ähnliche Antworten
- Finden und Ausnutzung dieser Ähnlichkeiten zur Generierung einer neuen Antwort

# Der 1. Ansatz (vereinfacht)

Q: "How tall is the Mount Everest?"  
A: [8848m] (in Form einer Entität)

Neue Frage mit Antwort als Entität

Bestimmung des  
Clusters über  
Ähnlichkeitsmaße



Sammlung  
Fragecluster

Cluster mit  
Sammlung  
Frage-Antwort  
Paaren

Q: "How tall is the Eiffeltower?"  
A: "The Eiffeltower is [ANS] tall"

Q: "How tall is Micheal Jordan?"  
A: "Micheal Jordan is [ANS] tall"

...

Allgemeine Antwort  
Durch Analyse vorhandener  
Antworten

A: "[ENT] is [ANS] tall"

Antwort auf Neue Frage

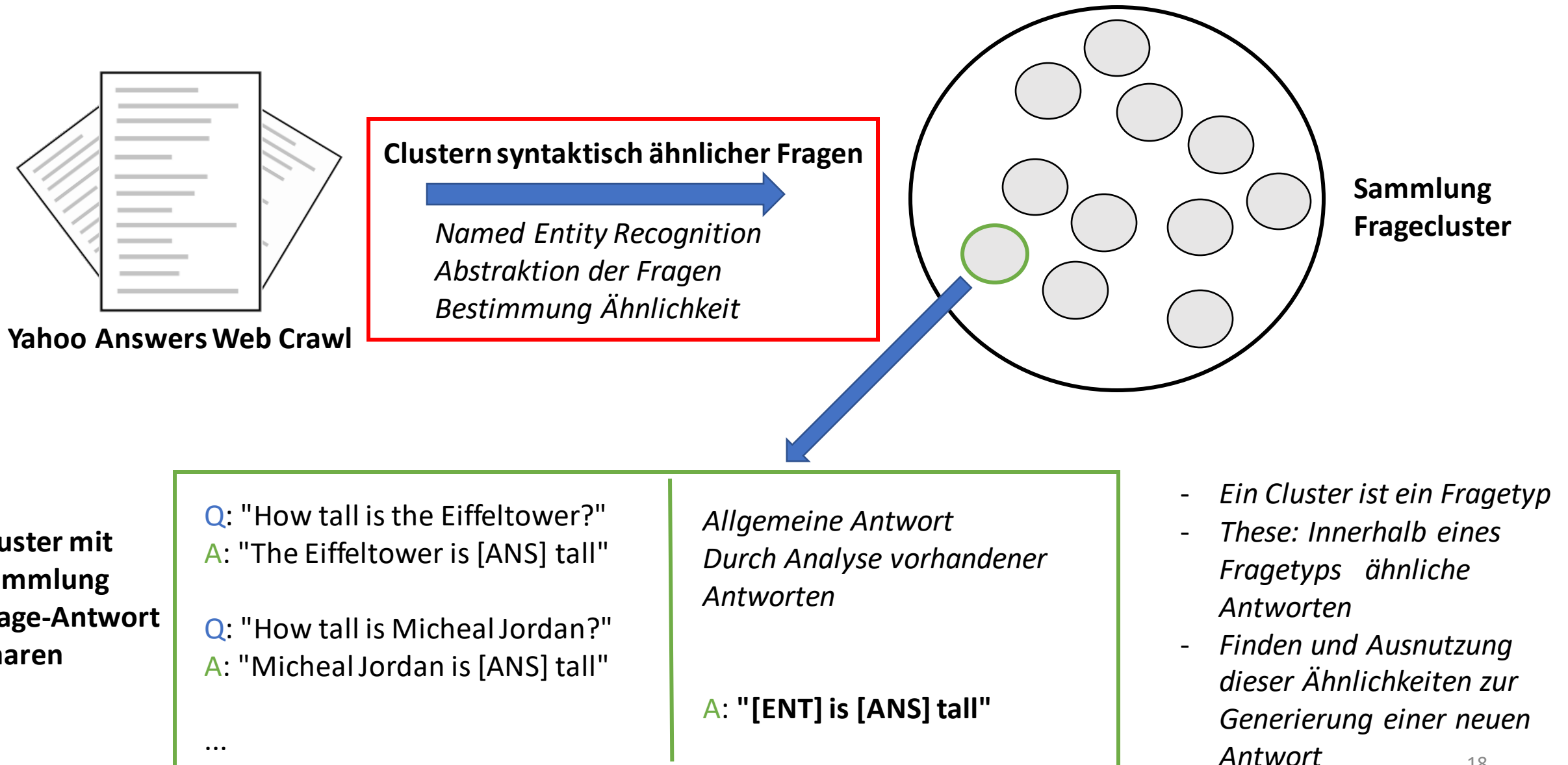
A: "The Mount Everest is  
[8848m] tall"



# Der 1. Ansatz

- Nutzung eines riesigen Frage-Antwort Datensatzes
  - Web Crawl von Yahoo Answers
  - User generated Content
  - Umfasst ungefähr 14 Millionen englische Fragen und deren Antworten
  - Riesige Datenmenge führt zu hinreichend großen Clustern?

# Der 1. Ansatz (vereinfacht)



# Der 1. Schritt

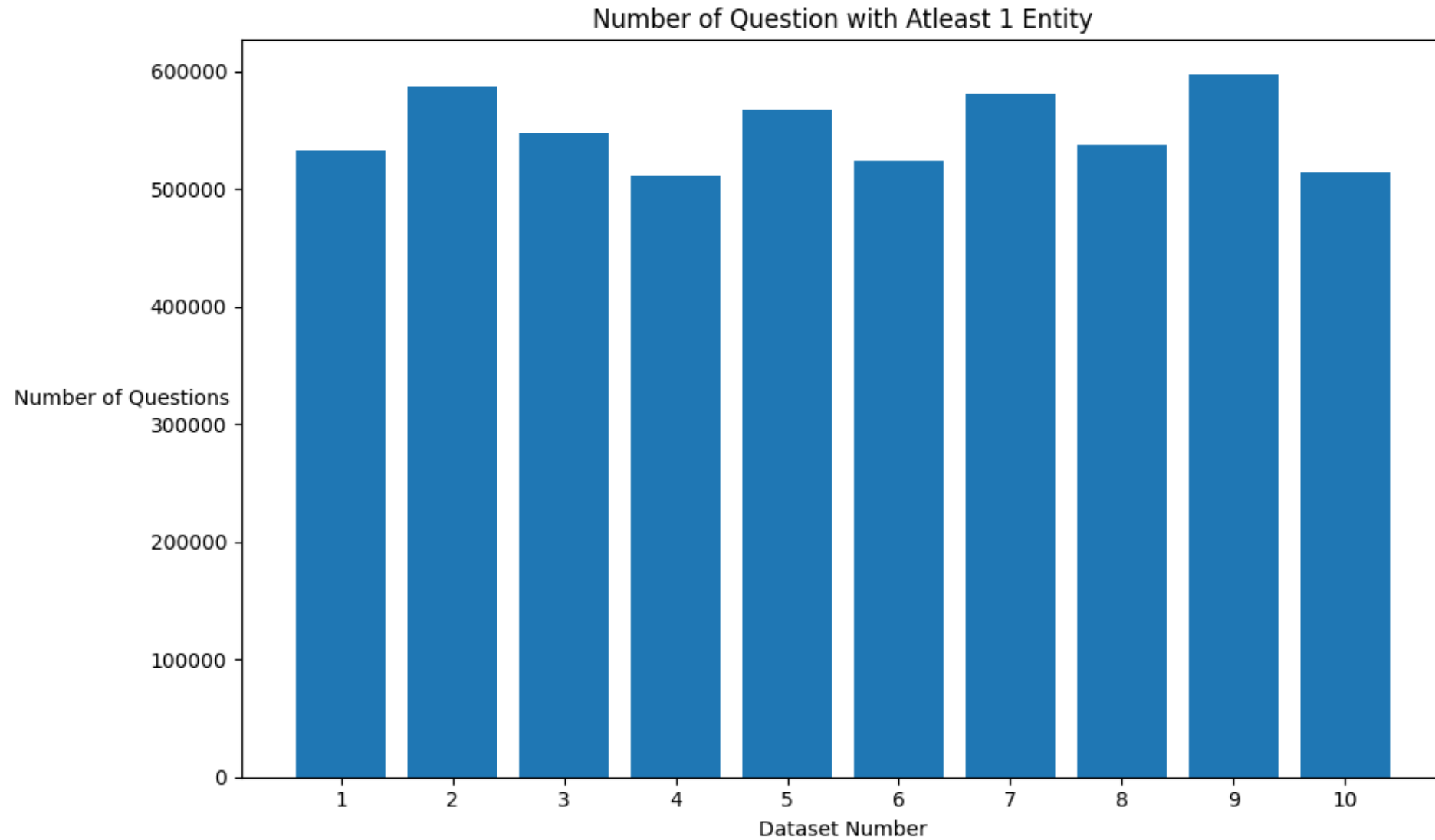
- Zuerst **Named Entity Recognition (NER)**
  - Finden und Ersetzen der Entitäten in den Fragen durch allgemeine Entitätstokens
  - Abstraktion der Fragen
  - Leichter Vergleich von Fragen und konzentrieren auf Struktur der Fragen
- Viele verschiedene Wege und Möglichkeiten dies zu tun
  - Entschieden für *TagMe*<sup>1</sup>
  - *Annotierung von Entitäten*

<sup>1</sup> <https://tagme.d4science.org/tagme/>

# Der 1. Schritt

- Datensatz mit den 14 Millionen englischen Fragen und Antworten in jeweils 10 kleinere Datensätze aufgesplittet
  - Aufgrund der enormen Größe und Zeitaufwand
- Für Jeden dieser Datensätze **NER** für die Fragen durchgeführt
  - Durchschnitt 550.000 Fragen mit Entitäten
  - Fragen mit Entitäten häufiger von QA beantwortbar
    - Auffindung vorhandener Entitäten im Graphen/Datenbank
    - Ansonsten keine faktischen Fragen
  - Beispiel: "Why is she not texting back?"

# Anzahl Fragen mit Entitäten



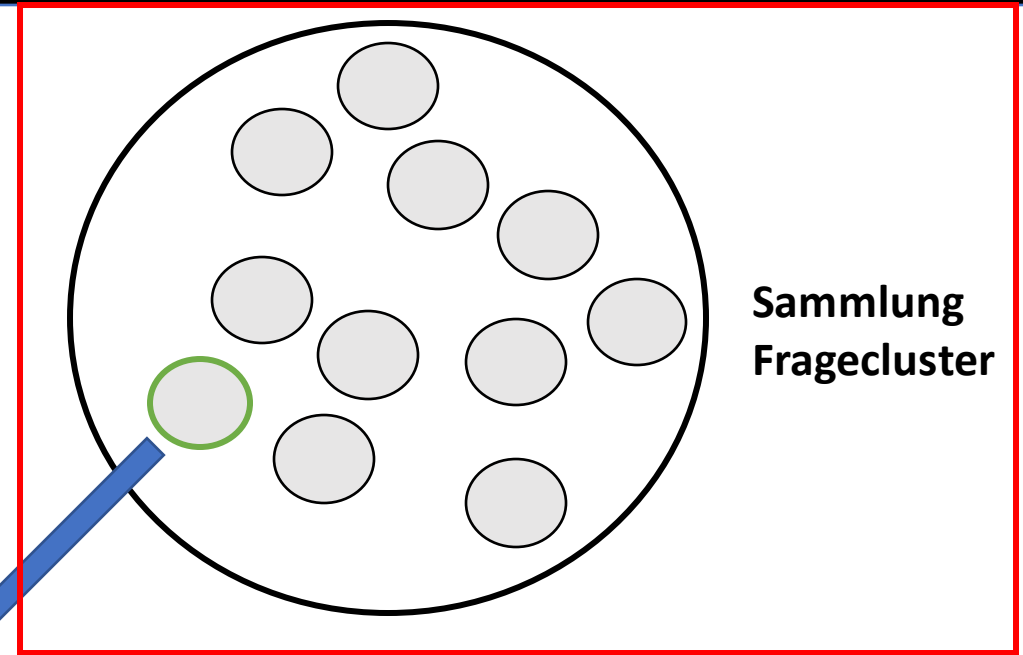
# Der 1. Ansatz (vereinfacht)



Clustern syntaktisch ähnlicher Fragen



*Named Entity Recognition*  
*Abstraktion der Fragen*  
*Bestimmung Ähnlichkeit*



Cluster mit Sammlung Frage-Antwort Paaren

Q: "How tall is the Eiffeltower?" A: "The Eiffeltower is [ANS] tall"	Allgemeine Antwort Durch Analyse vorhandener Antworten
Q: "How tall is Micheal Jordan?" A: "Micheal Jordan is [ANS] tall"	
...	A: "[ENT] is [ANS] tall"

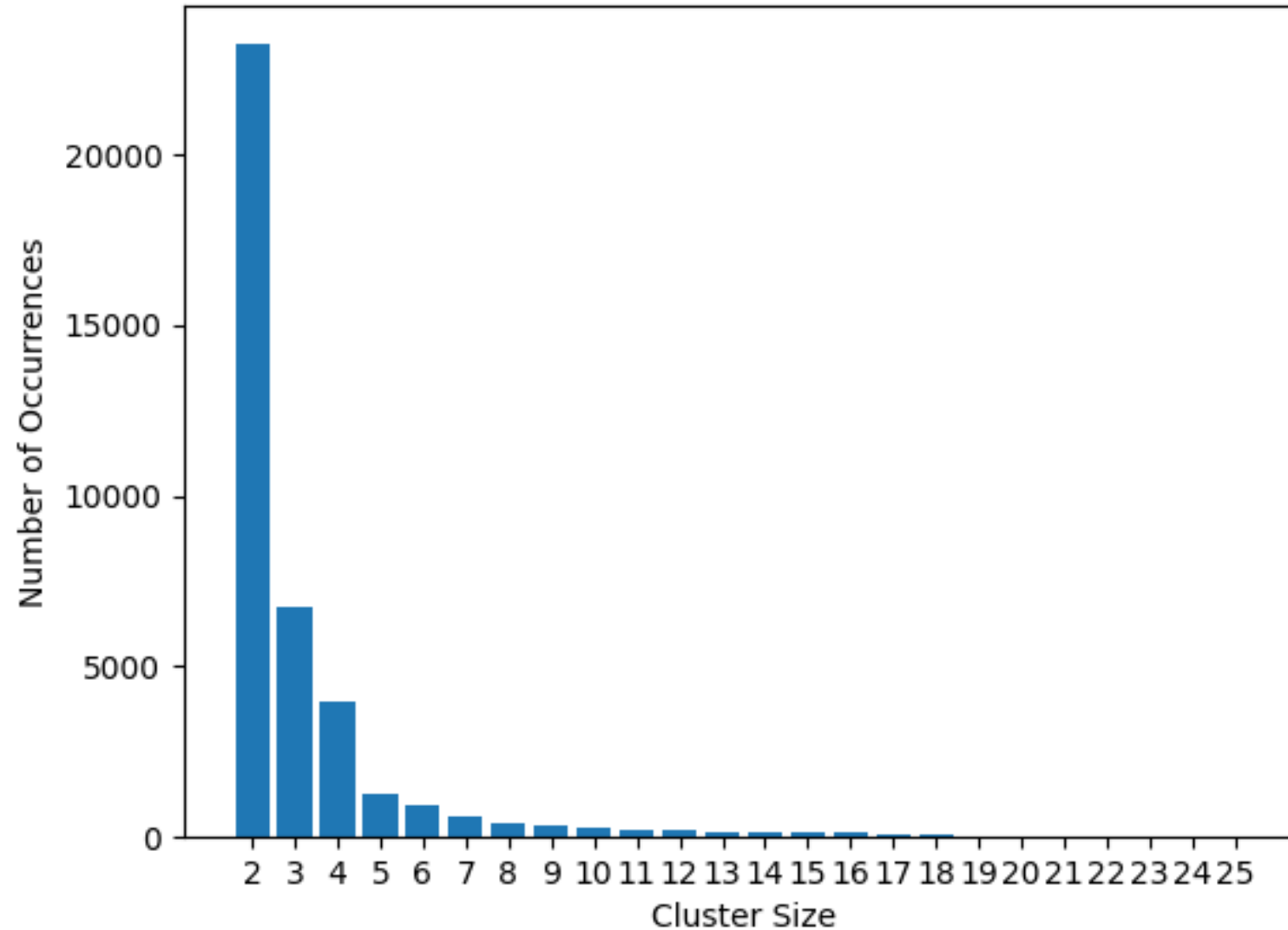
- Ein Cluster ist ein Fragetyp
- These: Innerhalb eines Fragetyps ähnliche Antworten
- Finden und Ausnutzung dieser Ähnlichkeiten zur Generierung einer neuen Antwort

# Der 2. Schritt

- Bildung der Fragecluster
  - Syntaktisch ähnliche Fragen werden zusammengefasst
  - 1 Cluster somit ein bestimmter Typ von Frage
- Für jede Frage innerhalb des Datensatzes werden mögliche Kandidaten über **Locality-Sensitive Hashing (LSH)** gefunden
  - Somit Sammlung von möglichen ähnlichen Kandidaten
  - Daraufhin Berechnung von **BLEU-Score** (*Papineni et al., 2002*) (Bestimmung Ähnlichkeit zwischen Texten über nGramme)
  - Es ergaben sich über diese Wege im Durchschnitt 40000 Cluster mit ähnlichen Fragen

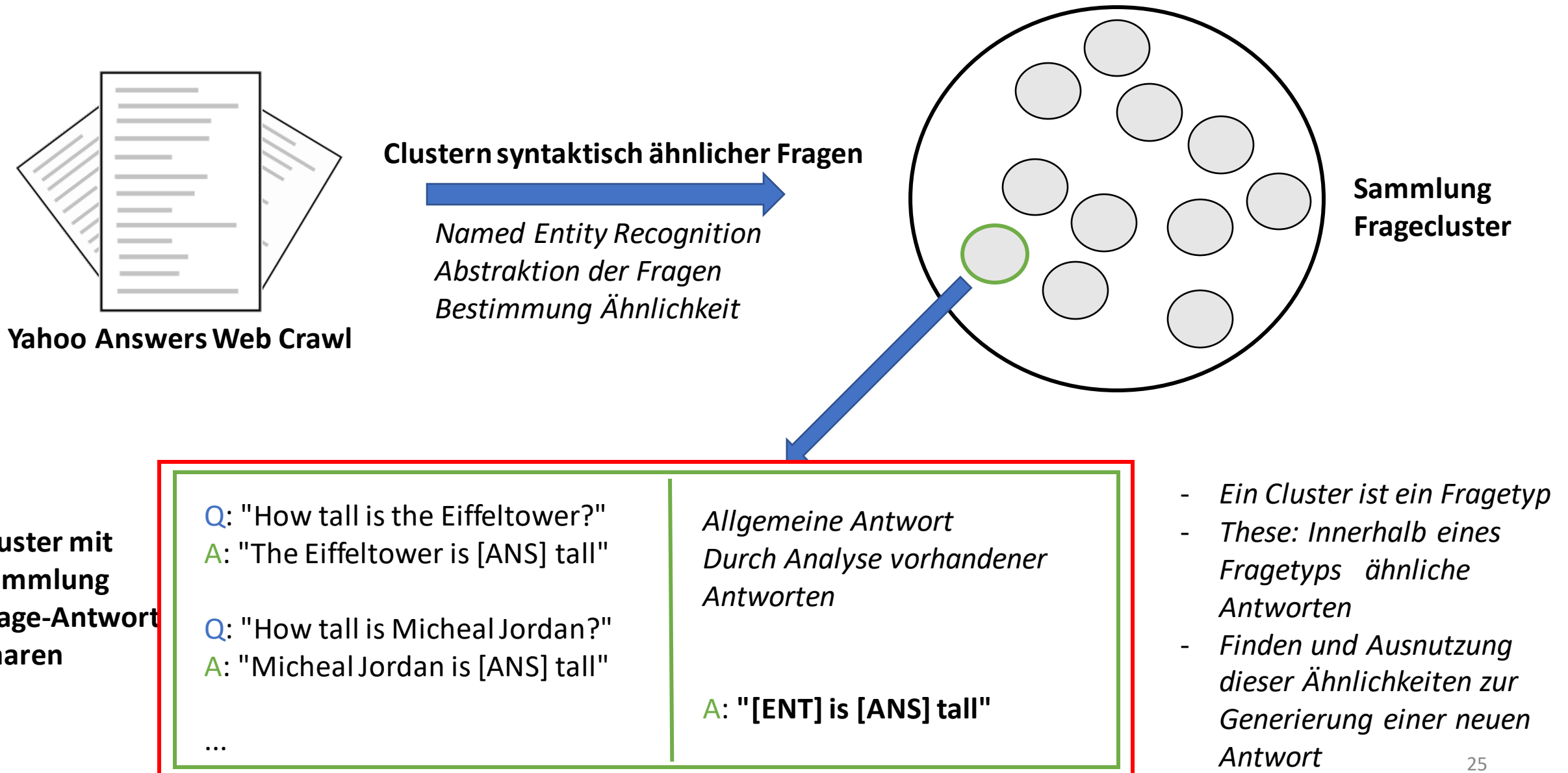
# Der 2. Schritt

Wie oft kommt ein Cluster der Größe N vor (40029 Cluster)





# Der 1. Ansatz (vereinfacht)



# Der 3. Schritt: Analyse der Antworten auf Ähnlichkeiten

- Cluster wurden gebildet nun Analyse der Antworten
  - Ebenfalls Finden und Ersetzen der Entitäten in den Antworten für Abstraktion
- Untersuchte Eigenschaften
  - **Ngramme** – *Unigramme, Bigramme, Trigramme, etc...*
    - Mögliche Ähnliche Satzstrukturen, die sich oft wiederfinden lassen
  - **Satzanfang / Satzenden**
    - Beginnen die Antworten normalerweise gleich, enden sie ähnlich
    - Ähnliche Strukturen am Anfang oder Ende?
  - **Anzahl der Wörter**
    - Wie lange ist normalerweise die Antwort
  - **Nutzung von dem 1. Antwortsatz, später Nutzung von der kompletten zusammengefassten Antwort**
    - Gibt es da Unterschiede von den Ergebnissen?
  - **Gibt es bestimmte Wörter (abgesehen von Stopwörtern) die oft vorkommen?**

# Der 3. Schritt: Analyse der Antworten auf Ähnlichkeiten - Fazit

- Enttäuschende Ergebnisse
  - Selten (<1%) Ähnlichkeiten zwischen den Antworten innerhalb eines Clusters
  - Selbes Ergebnis sowohl für Nutzung des ersten Antwortsatzes als auch für die Nutzung der kompletten Antwort
  - Dieses Experiment unterstützt nicht die These
- Mögliche Ursachen
  - Es sind immer noch usergenerierte Daten
    - Oft unsauber formatiert
    - Manchmal falsche, kurze, nichtssagende oder humorvolle Antworten als beste Antworten gewählt
    - Rechtschreibfehler
    - ungewöhnliche Formulierungen der Antworten

# Die 2. Idee

- **Idee 2.** Nutzung von Vorgefertigte Templates für Verbalizations
- Datensatz: **VQuAnDa** (*Kacupaj et al., 2020*)
  - Sammlung von 5000 annotierten Fragen mit verbalisierten Antworten
  - Entstanden aus dem Datensatz **Lc-QuAD** (*Dubey et al. 2019*)
    - Gold Standard Dataset für Question Answering over Knowledge Base mit 5000 Fragen
    - Ergab sich aus 42 SPARQL Queries (42 Fragetypen) über Knowledge Base *DBPedia*<sup>2</sup>
  - VQuAnDa hat zu diesem Datensatz eine ordentliche Verbalisierung der Antworten hinzugefügt

<sup>2</sup> <https://www.dbpedia.org/resources/>

# Die 2. Idee



Lc-Quad

+



VQuAnDa



Neuer Datensatz (Lc-Quad + VQuAnDa)

# Die 2. Idee



Lc-Quad

+



VQuAnDa



## Question

"What is the allegiance of John Kotelawala?"

## Question Entities

"What is the <allegiance> of <John Kotelawala>?"

## Abstract Question

"What is the [ENT] of [ENT] ?"

"What is the [0] of [1] ?"

## Answer

"The allegiance of John Kotelawala is [Sri Lanka]."

## Abstract Answer

"The [0] of [1] is [ANS]."

Beispiel eines Eintrages des kombinierten Datensatzes

# Die 2. Idee



# Die 2. Idee

## Neue Frage:

"What is the nationality of Angela Merkel?"

**Antwort:** [German]

## Question

"What is the allegiance of John Kotelawala?"

## Question Entities

"What is the <allegiance> of <John Kotelawala>?"

## Abstract Question

"What is the [ENT] of [ENT] ?"

"What is the [0] of [1] ?"

## Answer

"The allegiance of John Kotelawala is [Sri Lanka]."

## Abstract Answer

"The [0] of [1] is [ANS]."

Beispiel einer Frage aus dem Datensatz N



# Die 2. Idee

## Neue Frage:

"What is the nationality of Angela Merkel?"

## Abstrahiert:

"What is the [ENT] of [ENT]?"

Antwort: [German]

## Question

"What is the allegiance of John Kotelawala?"

## Question Entities

"What is the <allegiance> of <John Kotelawala>?"

## Abstract Question

"What is the [ENT] of [ENT] ?"

"What is the [0] of [1] ?"

## Answer

"The allegiance of John Kotelawala is [Sri Lanka]."

## Abstract Answer

"The [0] of [1] is [ANS]."

Beispiel einer Frage aus dem Datensatz N

# Die 2. Idee

**Nutzung der Abstrakten  
Antwort und Einsetzung  
der Entitäten aus der  
Neuen Frage**

**Neue Frage:**

"What is the **nationality**  
of **Angela Merkel**?"

**Abstrahiert:**

"What is the **[ENT]** of  
**[ENT]**?"

**Antwort:** [German]

**Question**

"What is the allegiance of John Kotelawala?"

**Question Entities**

"What is the **<allegiance>** of **<John Kotelawala>**?"

**Abstract Question**

"What is the **[ENT]** of **[ENT]** ?"

"What is the **[0]** of **[1]** ?"

**Answer**

"The **allegiance** of **John Kotelawala** is [Sri Lanka]."

**Abstract Answer**

"The **[0]** of **[1]** is **[ANS]**."

**Beispiel einer Frage aus dem Datensatz N**

# Die 2. Idee

## Neue Frage:

"What is the **nationality** of **Angela Merkel**?"

## Abstrahiert:

"What is the **[ENT]** of **[ENT]**?"

**Antwort:** [German]

## Generierte Antwort:

The **nationality** of **Angela Merkel** is German

## Question

"What is the allegiance of John Kotelawala?"

## Question Entities

"What is the **<allegiance>** of **<John Kotelawala>**?"

## Abstract Question

"What is the **[ENT]** of **[ENT]** ?"

"What is the **[0]** of **[1]** ?"

## Answer

"The **allegiance** of **John Kotelawala** is [Sri Lanka]."

## Abstract Answer

"The **[0]** of **[1]** is [ANS]."

Beispiel einer Frage aus dem Datensatz N

# Die 2. Idee

- Somit 5000 Antworttemplates
- Nutzung der vorgefertigten Antworten zur Bildung neuer Antworten
  - Finden von ähnlichen Fragen aus dem Datensatz gleicher Ansatz wie Versuch 1
    - *Named Entity Recognition, Abstraktion, LSH + BLEU-Score*
  - Einsetzen der Entitäten in die Antwort
    - Position wo welche Entitäten stehen sind bekannt
  - Durchführung für jede gefundene ähnliche Fragen aus dem Datensatz
  - Grammatikalisch falsche Sätze werden verworfen
    - Rückgabe aller "richtige" Sätze

# Stand Idee 2

- Nutzung der vorgefertigten Antworten zur Bildung neuer Antworten
  - Finden von ähnlichen Fragen aus dem Datensatz gleicher Ansatz wie Versuch 1
    - *Named Entity Recognition, Abstraktion, LSH + BLEU-Score*
  - Einsetzen der Entitäten in die Antwort
    - Position wo welche Entitäten stehen sind bekannt
  - Durchführung für jede gefundene ähnliche Fragen aus dem Datensatz
  - Grammatikalisch falsche Sätze werden verworfen
    - Rückgabe aller "richtige" Sätze

**Fortschritt: ungefähr 90%**

- Näher untersuchen, an was meine These im ersten Versuch scheiterte
  - Wie könnte man das erste Experiment verändern (anderer Datensatz?, andere Herangehensweise?, etc...)
- Lässt sich die Verbalisierung von solchen Antworten durch einen Template-basierten Ansatz lösen (2. Ansatz)?
  - Würde ebenfalls These unterstützen, dass ähnliche Fragen zu ähnlichen Antworten führen
- Falls 2. Ansatz ebenfalls nicht funktioniert wie erwartet
  - Ist die These zu simpel? Führen ähnliche Fragen nicht zu ähnlichen Antworten?
  - Was könnten weitere Ansätze sein um dieses Problem zu lösen?

# Diskussion & Anregungen

- **Gibt es irgendwelche Ideen und Anregungen?**
- **An was habe ich vielleicht noch nicht gedacht?**
- **Fragen über die Folien?**

# Quellen Paper

- Endri Kapupaj, Shyamnath Premnadh, Kuldeep Singh, Jens Lehmann, and Maria Maleshkova. "VOGUE: Answer Verbalization through Multi-Task Learning", arXiv2106:13316 (2021)
- Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. "Complex Knowledge Base Question Answering: A Survey", arXiv:2108.06688 (2021)
- Endri Kacupaj, Kuldeep Singh, Maria Maleshkova, and Jens Lehmann. "An Answer Verbalization Dataset for Conversational Question Answerings over Knowledge Graphs", arXiv.2208.06734 (2022)
- Dubey, M., Banerjee, D., Abdelkawi, A., Lehmann, J. (2019). LC-QuAD 2.0: A Large Dataset for Complex Question Answering over Wikidata and DBpedia. In: , et al. The Semantic Web – ISWC 2019. ISWC 2019. Lecture Notes in Computer Science(), vol 11779. Springer, Cham. [https://doi.org/10.1007/978-3-030-30796-7\\_5](https://doi.org/10.1007/978-3-030-30796-7_5)
- Kacupaj, E., Zafar, H., Lehmann, J., Maleshkova, M. (2020). VQuAnDa: Verbalization QUestion ANswering DATaset. In: , et al. The Semantic Web. ESWC 2020. Lecture Notes in Computer Science(), vol 12123. Springer, Cham. [https://doi.org/10.1007/978-3-030-49461-2\\_31](https://doi.org/10.1007/978-3-030-49461-2_31)
- Papineni, Kishore & Roukos, Salim & Ward, Todd & Zhu, Wei Jing. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. 10.3115/1073083.1073135.



- Folie 3 und fortfolgend:
  - Zahnräder: <https://de.vecteezy.com/vektorkunst/552199-zahnrad-vektor-icon>
- Folie 12:
  - OpenAI Logo: <https://openai.com/>