

# **Schlussbericht zum Vorhaben „Verarbeiten und Beantworten natürlichsprachlicher Anfragen auf mobilen Endgeräten mit beschränkten Ressourcen“**

im Rahmen des Eurostarsprojektes  
**E!9725 QAMEL**  
„Fragebeantwortung auf mobilen Endgeräten“

Prof. Dr. Axel-Cyrille Ngonga Ngomo, Dr. Ricardo Usbeck,  
René Speck, Daniel Vollmers, Jan Reinecke, Nadine Jochimsen

Universität Leipzig, Universitätsrechenzentrum  
Augustusplatz 10, 04109 Leipzig

Gefördert durch das Bundesministerium für Bildung und  
Forschung (BMBF)

Förderkennzeichen: 01QE1549C

Projektlaufzeit: 01.11.2015 – 31.12.2018

## Inhaltsverzeichnis

<b>1</b>	<b>Kurze Darstellung</b>	<b>3</b>
1.1	Aufgabenstellung . . . . .	3
1.2	Voraussetzungen . . . . .	3
1.3	Planung und Ablauf des Vorhabens . . . . .	4
1.4	Wissenschaftlicher Stand, an den angeknüpft wurde . . . . .	4
1.5	Zusammenarbeit mit anderen Stellen . . . . .	5
<b>2</b>	<b>Eingehende Darstellung</b>	<b>6</b>
2.1	Verwendung der Zuwendung und erzielte Ergebnisse . . . . .	6
2.1.1	Verwendung der Zuwendung . . . . .	6
2.1.2	Erzielte Ergebnisse . . . . .	6
2.2	Zahlenmäßiger Nachweis . . . . .	19
2.3	Notwendigkeit und Angemessenheit der geleisteten Arbeit . . . . .	19
2.4	Voraussichtlicher Nutzen . . . . .	22
2.5	Fortschritt bei anderen Stellen . . . . .	23
2.5.1	Wissensextraktion . . . . .	23
2.5.2	Frageformulierung . . . . .	24
2.5.3	Fragebeantwortung . . . . .	24
2.5.4	Mobile RDF Triple Stores . . . . .	24
2.6	Erfolgte und geplante Veröffentlichungen . . . . .	25
2.6.1	Erfolgte Veröffentlichungen . . . . .	25
2.6.2	Geplante Veröffentlichungen . . . . .	27

## Abbildungsverzeichnis

1	Architektur von FOX . . . . .	8
2	Architektur von Ocelot für RE . . . . .	9
3	Screenshot mobile, persönliche Daten auf mobilem Endgerät . . . . .	17

## Tabellenverzeichnis

1	F-Maß für die token- und entitätenbasierte Evaluation auf fünf Datensätzen in Englisch im Vergleich zu den integrierten Werkzeugen. . . . .	9
2	Durchschnittliches F-Maß und durchschnittliche Genauigkeit auf mehreren mehrsprachigen Silberstandard-Datensätzen. . . . .	9
3	Durchschnittliches F-Maß und durchschnittliche Genauigkeit auf mehreren mehrsprachigen Goldstandard-Datensätzen. . . . .	10
4	Der Durchschnitt für Genauigkeit (P), Vollständigkeit (R) und F-Maß (F1) für die Relationen <code>spouse</code> , <code>birthPlace</code> , <code>deathPlace</code> und <code>subsidiary</code> für die k besten Muster. . . . .	10

# 1 Kurze Darstellung

## 1.1 Aufgabenstellung

Ziel von QAMEL war die Konzeption, Implementierung und Evaluation eines modularen Frage-Beantwortungs-Systems (engl: Question Answering System) für mobile Endgeräte. Im Gegensatz zu existierenden Lösungen (welche auf einer Client-Server-Architektur basieren) sollte in diesem Projekt entwickelte Technologie ebenfalls vollständig auf dem mobilen Endgerät lauffähig sein. Damit sollte sichergestellt werden, dass auch im Offline-Modus Antworten generiert werden können. Die Universität Leipzig setzte sich in diesem Rahmen mit der Erarbeitung von ressourcenschonenden Algorithmen zur Generierung von SPARQL Anfragen auseinander.

Insbesondere wurden folgende Kategorien von Methoden untersucht:

- Extraktion von strukturierten Daten (RDF) aus tabellarischen Daten;<sup>1</sup>
- Extraktion und Konvertierung von (Nutzer-)daten aus mobilen Endgeräten;
- Konvertierungsalgorithmen zur Überführung von (multilingualen) unstrukturierten Datenquellen in strukturierte Daten, d.h. RDF;<sup>2</sup>
- Anfragekorrigierung (engl. Query correction)
- Anfragekomplettierung (engl. Query completion)
- Anfragealternativen (engl. “Did you mean”)
- Question Answering auf mobilen Endgeräten (der OQA Algorithmus)
- Online Question Answering (der TeBaQA Algorithmus)
- Evaluation von Offline bzw. mobilen Triple Stores
- Erprobung der Gesamtheit der Methoden in einem enzyklopädischen Anwendungsfall.

Alle 14 Deliverables können auf der Projektwebseite <https://qamel.eu/> eingesehen werden, eine Demo sowie QAMEL spezifischer Quellcode ist unter <https://github.com/dice-group/qamel> herunterladbar.

## 1.2 Voraussetzungen

Das Vorhaben wurde in 3 Jahren als Verbundprojekt durchgeführt. Die Leitung des Konsortiums übernahm die Firma Saltlux aus Korea. Weitere Projektpartner waren die Universität Leipzig (ULEI), das Korea Advanced Institute of Science and Technology (KAIST) und die bitstars GmbH aus Aachen. Die Projektlaufzeit wurde für die Universität Leipzig um 2 Monate kostenneutral verlängert.

---

<sup>1</sup><https://github.com/dice-group/TAIPAN>

<sup>2</sup><https://github.com/dice-group/Ocelot>, <https://github.com/dice-group/AGDISTIS>

### 1.3 Planung und Ablauf des Vorhabens

Das Vorhaben wurde in 9 Arbeitspaketen durchgeführt. Im Rahmen des ersten Arbeitspakets werden die Anforderungen von Endanwendern erhoben. Die technischen Arbeitspakete (2-5) setzen sich mit der Umsetzung dieser Anforderungen mittels einer modularen QA Architektur auseinander. Die darauf folgenden Anwendungsfälle (Arbeitspaket 6-8) dienen der Erprobung sowie Qualitätssicherung der entwickelten Lösung. In AP9 wird das Projektmanagement realisiert.

Die Verantwortlichkeit für AP 1-9 wurden entsprechend der gemeinsamen Vorhabensbeschreibung unter den Mitgliedern des Konsortiums verteilt. Monatliche virtuelle Treffen sowie mehrere Treffen bei einzelnen Projektpartnern erlaubten eine zeiteffiziente Zusammenarbeit.

Die Universität Leipzig erarbeitete/erweiterte die Frameworks FOX, AGDISTIS und TAIPAN, TeBaQA und OQA sowie ein Satz an Werkzeugen, welche für QAMEL speziell implementiert wurden. Desweiteren erarbeitete die Universität Leipzig diverse Evaluationsdatensätze und QA-Systeme für AP4, welche in das zusätzlich erweiterte **GERBIL-QA** Framework für Question-Answering-Systeme Eingang gefunden haben.<sup>3</sup> KAIST, Saltlux und bitstarts erweiterten bzw. implementierten ihre eigenen QAMEL-Tools und QAMEL-Datensätze. Die Gesamtheit der Werkzeuge floß in die kombinierten Anwendungsfälle (engl. Use Cases) ein, welche von allen Partnern unter Verwendung der implementierten Verfahren umgesetzt und evaluiert wurden. Im Rahmen des Projekts erreichte die Universität Leipzig alle im Antrag versprochenen Ziele. Die Ergebnisse der Universität Leipzig stehen als Erweiterungen der oben genannten Frameworks zur Verfügung und wurden in Publikationen auf internationalen Konferenzen beschrieben.

### 1.4 Wissenschaftlicher Stand, an den angeknüpft wurde

Zu Beginn des Projektes existierten die Frameworks FOX, AGDISTIS, NLIWOD<sup>4</sup> und HAWK.<sup>5</sup> Keines dieser Werkzeuge stellte die zu entwickelnden Funktionen zur Verfügung (siehe Abschnitt 1.1). Sie bildeten jedoch die technischen Grundlagen für die Implementierung von Teilen der entwickelten Verfahren.

Neue Frameworks wie GERBIL QA, TeBaQA und OQA entstanden während der Projektlaufzeit. Als Fachliteratur wurden die in den QAMEL Publikationen genannten Veröffentlichungen und Bücher verwendet. Als Informationsdienste dienten öffentliche Publikationsportale wie Google Scholar,<sup>6</sup> DBLP<sup>7</sup> und die ACM Digital Library<sup>8</sup>, der IEEE Explorer<sup>9</sup> sowie die Bibliothek der Universität Leipzig.

---

<sup>3</sup><http://gerbil-qa.aksw.org/gerbil/>

<sup>4</sup><https://github.com/dice-group/NLIWOD>

<sup>5</sup><https://github.com/dice-group/HAWK>

<sup>6</sup><http://scholar.google.com>

<sup>7</sup><http://dblp.uni-trier.de/>

<sup>8</sup><http://dl.acm.org/>

<sup>9</sup><https://ieeexplore.ieee.org>

## 1.5 Zusammenarbeit mit anderen Stellen

Im Rahmen des Projekts wurde der Austausch mit anderen Stellen gesucht. Insbesondere wurden Treffen mit Mitarbeitern von universitären Einrichtungen wie bspw. der Hochschule Beuth, der Universität Bonn, der Universität Halle-Wittenberg und der Universität Paderborn organisiert. Desweiteren wurde sich mit Mitarbeitern der EU-Projekte HOBBIT, DIESEL, BigDataEurope und WDAqua ausgetauscht. Ebenfalls nahm das Konsortium an der Koordinierung von mehreren Workshops auf internationalen Konferenzen teil, um den aktuellsten Stand der Technik in Erfahrung zu bringen und Entwicklungen dem Stand der Technik entsprechend vorantreiben zu können. In allen Fällen war das Ziel der Zusammenarbeit die Minimierung von doppelten Entwicklungen, eine erweiterte Dissemination und somit die Optimierung des Nutzen/Kosten-Verhältnisses von QAMEL.

## 2 Eingehende Darstellung

### 2.1 Verwendung der Zuwendung und erzielte Ergebnisse

#### 2.1.1 Verwendung der Zuwendung

Die Zuwendung wurde ausschließlich für nicht-wirtschaftliche und für das Vorhaben notwendige Zwecke genutzt. Hauptsächlich wurden die Mittel für Personalausgaben wissenschaftlicher Mitarbeiter/innen sowie für die Anschaffung von Geräten verwendet. Es wurden drei für die Evaluation der Software relevante Android Smartphones sowie ein Server erworben, auf dem die Entwicklung des gesamten Projekts (u.a. Implementierung der Lösung, Evaluation der Module, Ausführung der Anwendungsfälle, Code-Korrekturen im Live-Betrieb) stattfand sowie die Berechnung von Datensätzen und Benchmarks. Server und Smartphones werden nach der Projektlaufzeit ebenso ausschließlich für wissenschaftliche Zwecke genutzt. Sachausgaben für Dienstreisen beliefen sich letztlich auf nur 0,7% der Personalausgaben und fielen damit geringer als ursprünglich geplant aus (siehe auch Kap. 2.2)

#### 2.1.2 Erzielte Ergebnisse

Die von ULEI erzielten Ergebnisse lassen sich in sechs Kategorien unterteilen:

1. eine Architektur für Verarbeitung und Beantwortung natürlichsprachlicher Anfragen auf mobilen Geräten,
2. mehrere Erweiterung von Frameworks für die Extraktion strukturierten Daten aus un- bzw. semi-strukturierten Datenquellen,
3. mehrere Systemprototypen zur Frageformulierung,
4. zwei Fragebeantwortungssysteme für den Online- und Offline-Betrieb,
5. die Auswertung von verfügbaren Triple-Stores zum Betrieb auf mobilen Endgeräten,
6. und ein Ansammlung von interaktiven Demos, Evaluationen und Anwendungsstudien.

Im Folgenden wird auf die in jedem dieser Bereiche erzielten Ergebnisse eingegangen. Anschließend werden die erzielten Ergebnisse mit den vorgegebenen Zielen verglichen.

**QAMEL Architektur** Ziel von QAMEL ist es, ein ressourcenbewusstes und generisches multimodales Framework zur Beantwortung von Fragen für mobile Geräte zu entwickeln, welches sowohl textuelle und sprachgetriebene Eingaben verarbeiten kann. Des Weiteren wird durch Verteilung der Daten neben einem Online-Modus, welcher eine Internetverbindung benötigt auch ein Offline-Modus implementiert, sodass Endanwender

das System auch ohne Internet nutzen können. Neben multimodalem Question Answering wird eine prototypische Erweiterungen um die von den Projektpartnern entwickelte Use-Cases entwickelt. Die Architektur von QAMEL besteht (1) aus einem generischen Skelett, welches Schnittstellen zu Nutzeroberflächen und Datenendpunkten bereitstellt. (2) Der Identifikation existierender Technologien, welcher zur Implementierung einiger Module verwendet werden könne. (3) Dem Design der Interaktion zwischen den einzelnen Komponenten basierend auf der entwickelten Architektur und (4) Der Integration von QAMEL in die Produkte der einzelnen Projektpartner, sowie der Generierung von Benchmarks für die einzelnen Module. Zum Test des Systems wurden drei Use-Cases definiert: (1) Suche in enzyklopädischen Daten, (2) Verbesserung des Nutzererlebnisses bei der Nutzung von Virtual-Reality und Augmented-Reality Systemen durch Question Answering und (3) ein smarterer Assistent zur Unterstützung von Touristen bei Sportereignissen in Korea.

**Erweiterung von Frameworks für die Extraktion strukturierten Daten aus un- bzw. semi-strukturierten Datenquellen** Die Extraktion von strukturierten Wissen aus unstrukturierten Datenquellen bildete den Kern von AP 2 und besteht im Wesentlichen aus drei Schritten: der Identifizierung von Entitäten (engl. Named Entity Recognition, kurz NER), der Disambiguierung der gefundenen Entitäten (engl. Named Entity Disambiguation, kurz NED) und der Extraktion von Relationen (engl. Relation Extraction, kurz RE) aus Text.

*Die Aufgabe der ULEI im Rahmen von AP2 war die Entwicklung eines auf DBpedia basierenden Wissensextraktions Frameworks für Fragen sowie Online-Texte.* Dieses Framework, das Federated Knowledge Extraction Framework (kurz FOX), basiert auf einer Vielzahl von Extraktionsalgorithmen sowie maschinellem Lernen (insbesondere Ensemble Learning) und nutzt neuronale Netze um Ergebnisse existierender NER Extraktionsalgorithmen miteinander zu kombinieren um bessere Ergebnisse als ein Einzelner zu liefern. Insbesondere werden durch das maschinelle Lernen die Stärken der einzelnen Extraktionsalgorithmen hervorgehoben und die Schwächen herabgesetzt. Zusätzlich zu der NER Funktionalität integriert FOX auch NED Verfahren zur Abbildung auf DBpedia Ressourcen. Schnittstellen für die Extraktion von expliziten Beziehungen zwischen Ressourcen, RE, in unstrukturierten Daten wurden ebenso implementiert.

Die Architektur von FOX (siehe Abbildung 1) besteht aus drei Schichten:

1. Die ML-Schicht implementiert Schnittstellen für überwachte maschinelle Lernverfahren. Hierbei gibt es zwei Phasen, eine Trainings- und eine Anwendungsphase. Zur Zeit sind 15 Lernverfahren implementiert. Die Evaluationen haben gezeigt, dass das feed-forward neuronale Netz die besten Ergebnisse liefert und wird somit aktuell verwendet.
2. Die Controller-Schicht implementiert die für den Einsatz des Frameworks notwendigen Kontrollflüsse. Jede Nutzereingabe wird zunächst gereinigt (entfernen von HTML und XML Tags sowie von unbekanntem Zeichen). Die Eingabe wird dann an alle vom Nutzer ausgewählten Extraktionswerkzeuge weitergeleitet. Die Ergebnisse

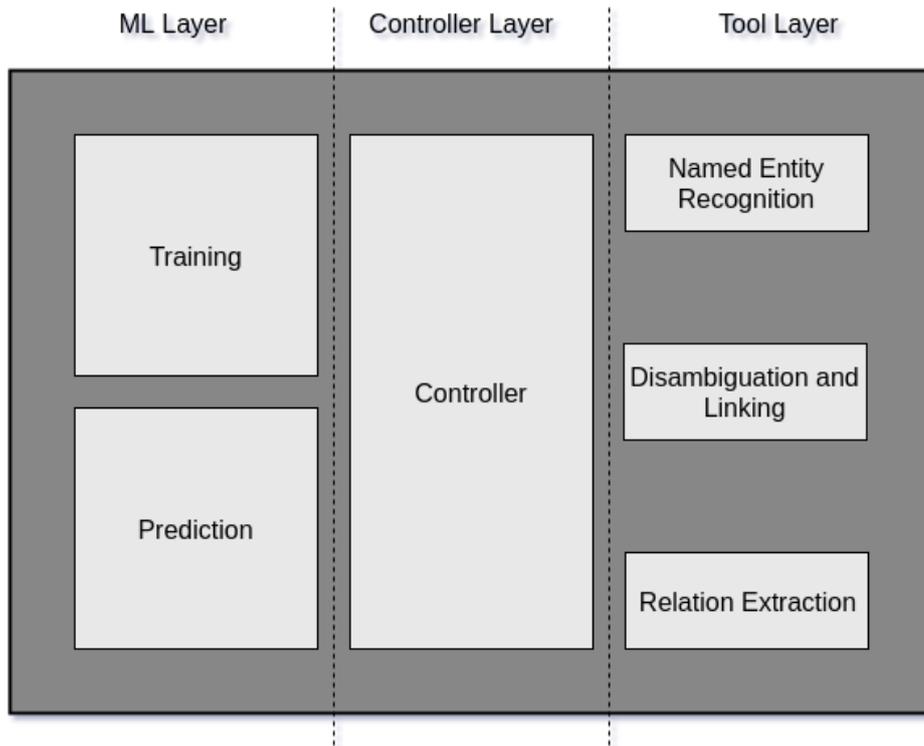


Abbildung 1: Architektur von FOX

dieser Extraktionswerkzeuge werden mit Hilfe der implementierten Lernverfahren in der ML-Schicht miteinander kombiniert. Anschließend wird die Ausgabe generiert und als RDF Serialisation formatiert.

3. Die Tool-Schicht stellt Schnittstellen für die NER, NED und RE Extraktionswerkzeuge zur Verfügung. Folgende Werkzeuge finden Verwendung in FOX.
  - a) Werkzeuge für NER: StanfordCoreNLP [11, 12, 18]<sup>10</sup>, Illinois [30]<sup>11</sup>, OpenNLP [2]<sup>12</sup>, Balie [24]<sup>13</sup> und Spotlight [6]<sup>14</sup>.
  - b) Werkzeuge für NED: AGDISTIS [50, 51] und MAG [20, 21].
  - c) Werkzeuge für RE: Boa [13], Patty [25], Ocelot [38] und StanfordCoreNLP [19] mit [31].

Die vorgestellten Evaluationen [29, 36, 37] der von FOX erzielten Ergebnisse zeigen, dass unser Framework vielen existierenden Lösungen überlegen ist. Ein Auszug aus den Evaluationen ist in Tabellen 1 bis 3 zu sehen. Das FOX Framework erreicht im besten

<sup>10</sup><http://nlp.stanford.edu:8080/ner/process>

<sup>11</sup>[http://cogcomp.cs.illinois.edu/page/demo\\_view/ner](http://cogcomp.cs.illinois.edu/page/demo_view/ner)

<sup>12</sup><http://opennlp.apache.org/download.html>

<sup>13</sup><http://balie.sourceforge.net/>

<sup>14</sup><http://spotlight.sztaki.hu/downloads/>

## 2 Eingehende Darstellung

Fall unserer Evaluation ein F-Maß von 95% auf einem Referenzkorpus besteht aus langen Nachrichten- und enzyklopädischen Texten.

Tabelle 1: F-Maß für die token- und entitätenbasierte Evaluation auf fünf Datensätzen in Englisch im Vergleich zu den integrierten Werkzeugen.

	token-based					entity-based				
	News	News*	Web	Reuters	All	News	News*	Web	Reuters	All
FOX	<b>92.73</b>	<b>95.23</b>	<b>68.81</b>	<b>87.55</b>	<b>90.99</b>	<b>90.70</b>	<b>93.09</b>	<b>63.36</b>	81.98	<b>90.28</b>
StanfordCoreNLP	90.34	91.68	65.81	82.85	89.21	87.66	89.72	62.83	79.68	88.05
Illinois	80.20	84.95	64.44	85.35	79.54	76.71	83.34	54.25	<b>83.74</b>	76.25
OpenNLP	73.71	79.57	49.18	73.96	72.65	67.89	75.78	43.99	72.89	67.66
Balie	71.54	79.80	40.15	64.78	69.40	69.66	80.48	35.07	68.71	67.82

Die von der ULEI implementierte Lösung für die Extraktion von Relationen, genannt Ocelot, basiert auf Syntaxbaummuster die generalisiert wurden um höhere Skalierbarkeit zu gewährleisten. Zusätzlich wurden weitere System für RE in FOX integriert.

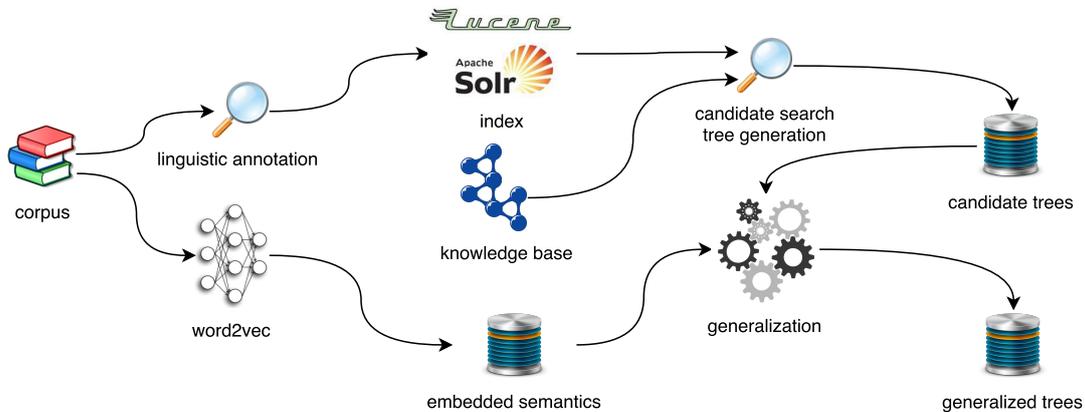


Abbildung 2: Architektur von Ocelot für RE

Ein Überblick des Datenflusses von Ocelot ist in Abbildung 2 dargestellt. Das System besteht im Wesentlichen aus vier Komponenten:

1. Linguistische Annotation: Hier werden die Ausgangsdaten, aktuell Wikipedia, mit Hilfe von DBpedia Ressourcen annotiert, um automatisch Trainingsdaten zu generieren.

Tabelle 2: Durchschnittliches F-Maß und durchschnittliche Genauigkeit auf mehreren mehrsprachigen Silberstandard-Datensätzen.

dataset	Balie	Illinois	OpenNLP	Spotlight	StanfordCoreNLP	FOX
	$F1-Score_T/pre_T$	$F1-Score_T/pre_T$	$F1-Score_T/pre_T$	$F1-Score_T/pre_T$	$F1-Score_T/pre_T$	$F1-Score_T/pre_T$
DE	35.91/50.88	-	-	34.06/ <b>79.17</b>	61.33/74.20	<b>63.00</b> /74.46
EN	56.23/64.87	70.57/70.14	46.30/58.53	57.22/74.21	76.70/78.61	<b>79.01</b> / <b>81.33</b>
ES	38.71/63.02	-	35.80/45.57	30.75/34.42	49.88/50.13	<b>64.57</b> / <b>74.58</b>
FR	47.12/71.53	-	58.40/86.01	58.48/ <b>87.97</b>	-	<b>71.90</b> /82.95
NL	-	-	49.41/ <b>79.96</b>	48.12/75.12	-	<b>65.41</b> /79.91

Tabelle 3: Durchschnittliches F-Maß und durchschnittliche Genauigkeit auf mehreren mehrsprachigen Goldstandard-Datensätzen.

dataset	Balie	OpenNLP	Spotlight	StanfordCoreNLP	FOX
	$F1-Score_T/pre_T$	$F1-Score_T/pre_T$	$F1-Score_T/pre_T$	$F1-Score_T/pre_T$	$F1-Score_T/pre_T$
<i>testa ES</i>	42.67/61.00	56.57/70.34	13.03/17.54	68.12/65.20	<b>74.26/74.70</b>
<i>testb ES</i>	43.59/65.09	64.73/73.17	21.97/50.04	59.16/68.98	<b>76.26/76.53</b>
<i>train ES</i>	38.53/58.66	72.53/72.65	28.60/28.30	66.58/63.96	<b>77.61/77.35</b>
<i>testa NL</i>	-	57.26/79.09	21.49/66.24	-	<b>59.67/82.02</b>
<i>testb NL</i>	-	60.46/ <b>77.75</b>	39.27/71.92	-	<b>63.28/71.57</b>
<i>train NL</i>	-	<b>70.85/74.11</b>	35.19/64.45	-	68.06/ <b>79.28</b>
<i>train DE</i>	28.33/37.70	-	35.34/76.00	45.97/53.69	<b>60.66/78.22</b>

2. Kandidatenauswahl: In dieser Komponente werden annotierte Sätze aus den Trainingsdaten ausgewählt, die sehr wahrscheinlich interessante Relationen zwischen Entitäten enthalten. Im Anschluss werden Syntaxbäume generiert und gespeichert.
3. Eingebettete Semantik: Hier wird Word2Vec mit Wikipedia verwendet um eingebettete Semantik zu erhalten, die später bei der Filterung der generierten und generalisierten Syntaxbaummuster helfen.
4. Generalisierung: In dieser Komponente werden die Syntaxbäume zu Syntaxbaummuster generalisiert und mit Hilfe der eingebettete Semantik gefiltert und zum Schluss die Muster bewertet.

Die Evaluationen der Ergebnisse von Ocelot [38] zeigen eine bessere Performanz gegenüber anderen existierenden Systemen (Boa und Patty) die auch in FOX integriert wurden. Ein Auszug der Evaluation ist in Tabelle 4 aufgeführt.

Tabelle 4: Der Durchschnitt für Genauigkeit (P), Vollständigkeit (R) und F-Maß (F1) für die Relationen `spouse`, `birthPlace`, `deathPlace` und `subsidiary` für die k besten Muster.

top k	Boa	Patty	Ocelot
	P/R/F1	P/R/F1	P/R/F1
1	75.00/8.120/14.58	75.00/9.550/16.67	<b>100.0/13.12/22.92</b>
2	62.50/12.66/20.94	62.50/15.39/24.24	<b>87.50/21.23/33.64</b>
3	58.33/18.51/27.86	66.67/24.94/35.36	<b>91.67/34.35/48.93</b>
4	56.25/23.05/32.42	62.50/29.48/38.99	<b>91.67/40.19/54.73</b>
5	60.00/32.60/41.46	60.00/34.03/42.29	<b>86.67/43.77/56.55</b>

Informationen zur Installation und Verwendung von FOX sind in der Dokumentation<sup>15</sup> zu finden. Es kann Docker, aber auch der Quellcode direkt verwendet werden.

<sup>15</sup><https://github.com/dice-group/FOX/tree/master/documentation>

## 2 Eingehende Darstellung

Eine Beispielanfrage mit Hilfe von Curl ist in Auflistung 2 aufgeführt. Die verwendeten Parameter in diesem Beispiel werden im JSON-Format in der Datei example.json übermittelt (Auflistung 3).

```
docker run -p 4444:4444 -it fox
```

Listing 1: Run with Docker.

```
curl -d "@example.json" -H "Content-Type: application/json; charset=utf-8" http://fox  
-demo.aksw.org/fox
```

Listing 2: Beispielanfrage

```
{  
  "input" : "Leibniz was born in Leipzig.",  
  "type"  : "text",  
  "task"  : "re",  
  "output": "turtle",  
  "lang"  : "en"  
}
```

Listing 3: Beispielparameter

**Systemprototypen zur Frageformulierung** Zur Frageformulierung wurden Module entwickelt, die lose gekoppelt sind und per REST angesprochen werden. Die Formulierung der Fragen erfolgt per Texteingabe oder per Sprache. Die Module dienen zur Vervollständigung von Fragen, der Korrektur, Erweiterung und Verbalisierung.

**Verbalisierung der Sprach- und Texteingaben** Die Spracheingabe ist die natürliche Weiterentwicklung der Sucheingabe und wird durch bestehende Standardtechnologien ermöglicht, die an die Bedürfnisse und Anwendungsfälle der Konsortialpartner angepasst wurden. Nach einer eingehenden Analyse und Bewertung wurde die Android-basierte Google API als Spracherkennungslösung ausgewählt, da sie offline, schnell, effizient und in allen drei benötigten Sprachen (d.h. für Deutsch, Englisch und Koreanisch) verfügbar ist. Beschreibungen und Source Code hier zu finden:

<https://github.com/dice-group/qamel> (mobile Version)

<https://github.com/dice-group/hawk-audio-input> (Serverversion)

**Fragenkorrektur** Dieses Modul beinhaltet eine Autokorrektur von Tippfehlern und Schreibfehlern. Der entwickelte Ansatz basiert auf einer neuartigen JSON-basierten REST-Schnittstelle. Auf mobilen Geräten werden modernste On-Device-Mechanismen genutzt, um einen minimalen Ressourcenverbrauch zu erreichen. Beschreibung und Source Code können hier gefunden werden:

<https://github.com/dice-group/qamel/tree/master/queryCorrection>

**Fragevervollständigung** Das Fragevervollständigungsmodul kann Suchanfragen mit Hilfe von maschinellem Lernen auf Ontologien und Logdaten vorschlagen. Sobald ein Nutzer beginnt eine Frage einzutippen werden bereits existierende Fragesegmente genutzt

um Suchanfragen vorzuschlagen, was eine schnellere Suche ermöglicht und dem Nutzer Suchanfragen vorschlägt, die das Frageantwortungssystem versteht. Das Modul ist weiter in der Lage Wörter zu vervollständigen, basierend auf strukturiertem Wissen wie es in DBpedia vorliegt. Dieser Ansatz ist serverbasiert. Für die mobile Version wurde eine Autovervollständigung integriert, die auf Daten, die auf dem mobilen Endgerät zu Verfügung stehen, basiert.

Für die Bewertung der Autovervollständigung wurde ein Datensatz von mcloud.de genutzt. ES wurde ein Lucene basierten Analysator auf dem Datensatz verwendet und ein Wert von 0,339 als Durchschnitt der Fläche unter der Kurve erreicht. Angesichts der geringen Anzahl von Anfragen und der Tatsache, dass die Anfrage bereits gesehen worden sein muss, um sie automatisch zu vervollständigen, kann diese Lösung ein großer Vorteil für zukünftige QAMEL-Erweiterungen sein. Beschreibung und Source-Code können hier gefunden werden:

<https://github.com/dice-group/qamel/tree/master/queryCompletion>

**Frageerweiterung "Meinten Sie dies-Feature/ Fragealternativen** Dieses Modul ermöglicht das Vorschlagen von ähnlichen Suchanfragen, wenn die gegebene Frage kein Ergebnis zurück liefert. Das Modul ist in der Lage einzelne Wörter der originalen Fragen basierend auf Word2Vec zu ersetzen. Dies kann serverseitig für jede individuelle Wissensbasis trainiert werden. (siehe <https://github.com/dice-group/Jword2vec>). Die Basic-Version diese Features funktioniert und wurde getestet. Wegen der zu geringen Anzahl von qualitativ hochwertigen Fragen je Use Case ist ein aktives Lernen jedoch nicht möglich. Beschreibung und Source-Code können hier gefunden werden:

<https://github.com/dice-group/qamel/tree/master/queryAlternatives>

Benchmark: In diesem Use Case wurde mit lexikalischen Daten gearbeitet. Hierfür wurde ein Benchmark-Datensatz mit mehr als 250 Fragen<sup>16</sup> erstellt. Zusätzlich wurden mit QALD-8<sup>17</sup> auf der Internationalen Semantic Web Konferenz mehr als 50 qualitativ hochwertige Fragen kreiert, die eine große Variation von Vorlagen und sprachlichen Aspekten abdecken. Mit Log-Daten aus HAWK<sup>18</sup> haben wir das "Meinten Sie dies" Feature getestet. Link zu Benchmark und User Story:

<https://github.com/dice-group/NLIWOD/tree/master/qa.datasets>

### **Fragebeantwortungssysteme für den Online- und Offline-Betrieb**

**Offline Fragebeantwortungssystem** Das Ziel des offline Fragebeantwortungssystems ist die Beantwortung von Nutzerfragen ohne einen leistungsfähigen Server. Da viele mobile Endgerät nur über stark begrenzte Ressourcen im Bezug auf Speicherplatz und CPU verfügen, wird ein QA System mit hoch effizienten Komponenten benötigt. Aus diesem Grund werden nur Fragen betrachtet, für deren Beantwortung nur eine binäre Relation benötigt wird, sodass immer nur ein Teil eines RDF-Triples unbekannt ist.

Das System basiert 1) auf einer semantischen Analyse der Frage, 2) der Klassifikation

---

<sup>16</sup><https://project-hobbit.eu/challenges/qald2017/>

<sup>17</sup><https://project-hobbit.eu/challenges/qald-8-challenge/>

<sup>18</sup>[https://github.com/RicardoUsbeck/PhD\\_Ricardo/tree/eratum/callingNumbers/hawk](https://github.com/RicardoUsbeck/PhD_Ricardo/tree/eratum/callingNumbers/hawk)

des Fragentyps zur Zuordnung eines Templates und 3) Zuordnung der gefundenen Ressourcen zu den offenen Variablen eines Templates. Insgesamt werden folgende 5 Schritte durchgeführt:

### 1. Filterung von Stopwörtern

Nach der Generierung von Tokens, anhand von Whitespaces, werden anhand einer Blacklist alle Token ausgefiltert, welche keine Bedeutung für die Beantwortung einer Frage haben. Die Blacklists sind dabei sprachabhängig, umfassen aber für jede Sprache nur einige wenige Token, sodass die Blacklists einfach durch einen Muttersprachler erstellt werden können.

### 2. Extraktion des Fragentyps

In diesem Schritt wird basierend auf dem Fragewort der Fragen- und Antworttyp ermittelt, da so die Zahl der Templatekandidaten stark reduziert werden kann. Die Liste der Fragewörter umfasst ebenfalls nur einige wenige Worte, jedoch wird in einigen Sprachen dieser Prozess durch Ambiguität erschwert.

### 3. Generierung von Kandidaten für Entitäten

Zur einfachen Generierung von Entitäten wird davon ausgegangen, dass entweder eine Entität oder eine Eigenschaft aus der Wissensbasis extrahiert werden muss, um die fehlenden Elemente des Templates zu füllen. Dazu wird für jedes verbleibende Token der Frage eine Sparql-Query basierend auf folgendem Template generiert:

```
SELECT DISTINCT ?x ?z WHERE ?x rdfs: label ?z . FILTER regex (str(?x),  
".* <WORD >.*") FILTER (lang (?z)='en ')
```

Dabei wird *<WORD >* jeweils durch das gesuchte Token ersetzt. Zur Anpassung an weitere Sprachen kann der Term *FILTER (lang (?z)='en ')* entsprechend angepasst werden.

### 4. Ranking der Kandidaten

Das Ergebnis der Query ist jeweils ein Tupel  $t = (W, E, L)$  mit  $W$  als Input Token,  $E$  als gefundene Entität und  $L$  des Labels der Entität. Die Kandidaten werden zunächst basierend auf der Menge der Token, denen die selbe Entität zugeordnet wurde gerankt. Falls zwei oder mehr Kandidaten gleich viele Token matchen, wird zusätzlich die Levensteindistanz für das Ranking genutzt. Falls anschließend immer noch Kandidaten gleich stark gerankt sind wird dem Kandidaten mit dem längsten Label die höchste Priorität zugeordnet.

### 5. Kandidaten Berechnung für Eigenschaften und Ranking

Abschließend wird versucht basierend auf dem ermitteltem Fragentyp und den zuvor gefundenen Entitäten Kandidaten für die Eigenschaften zu finden und zu Ranken. Die Literale der subject-property und property-object Paare haben jeweils bestimmte Datentypen, welche anhand des ermittelten Fragentyps zur Reduzierung der möglichen Eigenschaften verwendet werden können. Beginnend mit der am höchsten gerankten Entität werden alle Eigenschaften mit dem Entsprechendem Datentyp aus der Wissensbasis extrahiert. Anschließend wird erneut die Levenstein Distanz der Property-Labels mit den Wörtern der Frage berechnet. Eigenschaften, welche bereits im vorherigen Schritt erkannt wurden, erhöhen dabei den Reliabilitätswert für eine Eigenschaft. Die Liste der

Kandidaten wird anschließend nach diesem Reabilitätswert geordnet. Das endgültige Ergebnis wird abschließend durch Ausführung der resultierenden SPARQL-Query aus dem vollständig ausgeführtem Template ermittelt.

**Online Question Answering** Für den Server basierten Ansatz wurde das Question Answering Framework TeBaQA<sup>19</sup> entwickelt, welches auf einer Kombination aus SPARQL-Query Templates und maschinellem Lernen basiert. Die Templates werden automatisch durch Einsetzen von Placeholdern aus einem Trainingsdatensatz im QUALD Format extrahiert. Placeholder sind dabei entweder Prädikate, Entitäten oder Klassen. Zur Beantwortung neuer Fragen wird eine strukturell äquivalente Frage gesucht, zu welcher bereits eine SPARQL-Query existiert. Die Idee hierbei ist, dass durch strukturell ähnliche SPARQL-Queries strukturell ähnliche natürlich sprachliche Fragen beantwortet werden können.

Für das Training werden 220 Fragen des QUALD-8 und 215 Fragen des QUALD-7 Datensatzes verwendet, die hinsichtlich ihrer Komplexität stark variieren, da sowohl Fragen mit Vergleichen (größer als), Superlativen (der größte) und Aggregation (wie viele) enthalten sind. Neben der natürlich sprachlichen Frage, der SPARQL Query und der Antwort enthält der Datensatz für jede Frage zudem folgende Metadaten:

- **id**: Die eindeutige ID der Frage
- **answerType**: Der semantische Antworttyp (Resource, String, Number, Date, oder Boolean)
- **aggregation**: Wahrheitswert, der angibt, ob in der SPARQL-Query Aggregation verwendet wird.
- **onlyDbo**: Wahrheitswert, der angibt, ob ausschließlich Prädikate aus DBpedia verwendet werden.
- **hybrid**: Wahrheitswert der angibt, ob eine SPARQL Query sowohl auf unstrukturiertem Text als auch auf einem strukturierten RDF-Graphen basiert.

Zur Transformation einer natürlich sprachlichen Frage in eine SPARQL-Query müssen zunächst semantische Einheiten aus der Frage extrahiert und anschließend in das Query Template eingefügt werden. Da SPARQL eine Graph basierte Anfragesprache ist, bietet es sich an semantische Ähnlichkeit als Graphisomorphismus zu betrachten. Alle isomorphen Graphen werden einer Klasse mit einem Graph-Pattern für jedes Query-Template zugeordnet. Zur Zuordnung einer neuen Frage zu einer Klasse kann ein stochastisches Klassifikationsmodell verwendet werden. Dazu ist es notwendig eine Frage in einen Feature Vektor zu transformieren. Insgesamt werden sowohl semantische Features, welche inhaltliche Aspekte einer Frage repräsentieren als auch syntaktische Features, die die Struktur einer Frage abbilden verwendet. Zu den semantischen Features gehören:

- **QuestionWord (Nominal)**: Das Fragewort (z.B., Wer, Was, Gib).

---

<sup>19</sup><https://github.com/dice-group/TeBaQA>

## 2 Eingehende Darstellung

- EntityPerson (Bool): Gibt an, ob eine Person in der Frage erwähnt wird.
- NumberOfToken (Numerisch): Anzahl der Tokens exklusive Stores the number of tokens exclusive Satzzeichen.
- QueryResourceType (Nominal): Kategorisierung der Frage anhand einer Liste von Themen z.B.(Filme, Musik, Bücher)

Zu den syntaktischen Features gehören:

- Noun (Numerisch): Anzahl der Nomen.
- Number (Numerisch): Anzahl der numerischen Werte.
- Verb (Numerisch): Anzahl der Verben.
- Adjective (Numerisch): Anzahl der Adjektive.
- Comparative (Bool): Wahrheitswert, ob Komparative oder Adverbien in der Frage existieren.
- NamedEntities (Numerisch): Anzahl der Entitäten in der Frage.
- TripleCandidates (Numerisch): Anzahl der Triple, die notwendig sind, um eine Frage zu beantworten.

Die aus den Trainingsdaten extrahierten Fragevektoren werden für das Lernen eines Klassifikationsmodells verwendet. Zur Auswahl eines Klassifikationsmodells wurden verschiedene Modelle getestet. Das beste F-Measure erreichten Logit Boost(0,571) und Multilayer Perceptron(0,567). Allerdings war die Klassifikationsrate beim Multilayer Perceptron (0,902) deutlich höher als bei Logit Boost(0,741). Somit wurde für die Klassifikation der Fragen ein Multilayer Perceptron verwendet. Für das Trainieren und Evaluieren der Klassifikationsmodelle wurde das Java-Framework WEKA<sup>20</sup> eingesetzt, da es auch für nicht Domänen-Spezialisten leicht zu bedienen ist. Als Konfiguration wurde für jedes Modell die Standardkonfiguration verwendet. Für die Beantwortung einer Frage muss abschließend das durch den Klassifikationsalgorithmus identifizierte Template mit den gefundenen Klassen, Predikaten und Entitäten gefüllt werden.

Die Evaluation von TeBaQA, welche unter <http://w3id.org/gerbil/qa/experiment?id=201805290000> abgerufen werden kann zeigt, dass TeBaQA ein F-Measure von 0,27 erreicht, bei einer Precision von 0,28 und einem Recall von 0,26. Eine weitere Fehleranalyse ergibt dass einfache Fragen wie *How much is the total population of the European Union?* in den meisten Fällen korrekt beantwortet werden können. Komplexere Fragen wie *What other books have been written by the author of The Fault in Our Stars?* sind jedoch schwerer zu beantworten, da ihre Struktur sehr viel umfassender ist.

---

<sup>20</sup><https://www.cs.waikato.ac.nz/ml/weka/>

### **Auswertung von verfügbaren Triple-Stores zum Betrieb auf mobilen Endgeräten**

Zur Auswahl eines geeigneten Triple Stores wurden verschiedene Systeme basierend auf verschiedenen Merkmalen untersucht. Zu den Merkmalen gehören vor allem die verwendete Lizenz, das letzte Updatedatum und die verwendete Programmiersprache. Triple Stores mit unklarer Lizenzierung wie  $\mu$ Jena<sup>21</sup> oder  $\mu$ OR<sup>22</sup> wurden dabei ausgeschlossen. Des Weiteren wird ein Triple Store benötigt der auf Java basiert, da andere Sprachen nicht kompatibel mit der internen Struktur und des Code-Standards unseres QA-Systems sind. Am Ende dieses Prozesses wurde RDF4J<sup>23</sup> als Triplestore ausgewählt, da dieser als einziges System alle Kriterien erfüllt.

### **Anwendungsstudie Enzyklopädische Wissensbasis**

Frage-Beantwortungssysteme sind, ohne Verbindung zu einem leistungsfähigen Server und der alleinigen Nutzung der Ressourcen eines mobilen Endgeräts, nicht in der Lage gesprochene oder geschriebene Fragen eines Nutzers zu beantworten. Um antworten zu können benötigt das System Informationen, auch wenn der Nutzer keine Verbindung zum Internet hat, wie z.B. in einem Fabrikgebäude oder in ländlichen Gegenden. Die meisten mobilen Endgeräte sind hinsichtlich ihrer Ressourcen wie Speicher oder Prozessor eingeschränkt. Für QAMEL wurden zwei Use Cases implementiert, die auf mit KAIST, ULEI, Saltlux und bitstars gemeinsam entwickelten Technologien fußen. Das sind zum einen eine Erweiterung von OQA, dem ersten offline Frage-Beantwortungsalgorithmus, welcher einen eigenen mobilen Datenspeicher bietet, einen einfachen aber effektiven multilingualen Algorithmus für die Transformierung der natürlichen Sprachen nach SPARQL sowie seinen Fokus auf einen geringen Ressourcenverbrauch legt. Der Source-Code ist unter folgender URL zu finden: <https://github.com/dice-group/qamel>

Zum anderen hat KAIST Module aus WP2 und WP5 in eine kombinierte Android App namens Mobile K-Box integriert. Der Zweck dieser App ist es, einfache Schnittstellen zu bieten, um die Funktionalitäten der WP2 und WP5 Module in mobilen Umgebungen nutzen zu können. Präsentationsfolien und Video zu den Funktionalitäten sind hier zu finden: [https://drive.google.com/open?id=1u09qDCEQv7dNv6\\_0ERFm5hiFSxzsBzRa](https://drive.google.com/open?id=1u09qDCEQv7dNv6_0ERFm5hiFSxzsBzRa)

**Features** Die Abbildung 3 zeigt, wie Daten vom mobilen Endgerät nach RDF Triple extrahiert und konvertiert werden. Der Prozess besteht hauptsächlich aus folgenden Schritten:

1. Extrahieren von Kontakten und Kalenderdaten des mobilen Endgeräts in die App als vCard und vCal.
2. Zerlegen der vCard und vCal in Zeichenketten um wichtige Informationen wie Name, Telefonnummer, Emailadresse, Ereignisname, Uhrzeit und Start- und Enddatum zu ermitteln.

---

<sup>21</sup>[http://poseidon.ws.dei.polimi.it/ca/?page\\_id=59](http://poseidon.ws.dei.polimi.it/ca/?page_id=59)

<sup>22</sup><http://publica.fraunhofer.de/documents/N-101812.html>

<sup>23</sup><http://rdf4j.org/>

3. Im letzten Schritt werden diese Zeichenketten mittels eines RDF Generators in RDF-Triple konvertiert und an den OfflineDataSynchronizer gesendet, womit die App dann in der Lage ist Fragen bezüglich der Nutzerdaten zu beantworten.

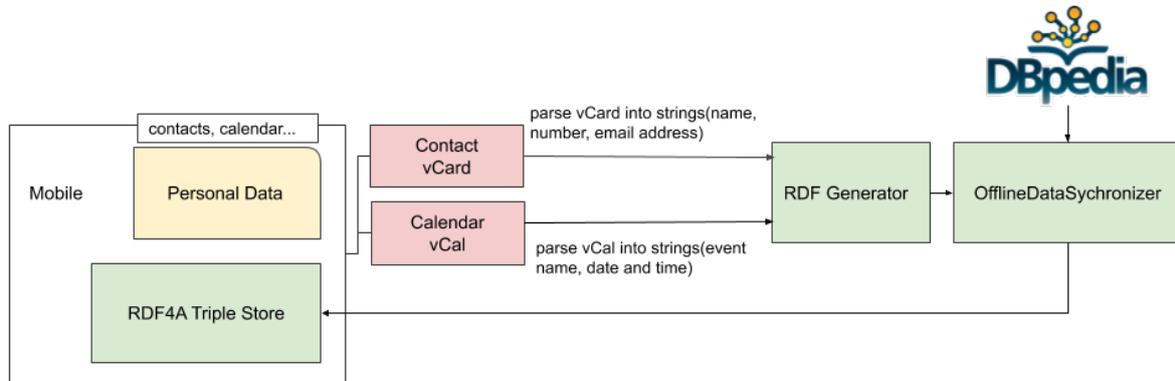


Abbildung 3: Screenshot mobile, persönliche Daten auf mobilem Endgerät

Damit die App<sup>24</sup> in der Lage ist, Fragen bezüglich der persönlichen Daten des Benutzers wie Kontakte, Kalenderereignisse und Ortsangaben zu beantworten, müssen die Nutzerdaten vom mobilen Enderät extrahiert werden. Durch Nutzung von Berechtigungen können Kontakte in vCards, Kalenderereignisse in vCal und Ortsangaben in Längen- und Breitengraden. Der Offline Data Builder ist allerdings nicht in der Lage VCard und vCal Formate zu lesen. Er kann nur RDF Triple lesen. So werden zuerst vCard und vCal in Zeichenketten zerlegt und dann benötigte Informationen wie Name, Telefonnummer, Emailadresse etc. daraus gewonnen. Anschließend nimmt der RDF Generator diese gewonnenen Informationen und generiert eine RDF-Datei im Turtle Format.

Der Offline Data Builder ist ein Werkzeug um offline Daten für die App zu generieren. Er liest Daten aus einer Datei. Für jedes Triple entscheidet er, ob die Daten relevante Informationen beinhalten oder nicht. Ein Triple wird als relevant angesehen, wenn alle drei Entitäten mindestens  $n$  (eine vorgegebene Anzahl) Vorkommen in der Ausgangsdatei vorweisen (ist das Objekt ein Literal: wenn die beiden anderen Entitäten diese Anzahl Vorkommen haben). Ein Triple wird gespeichert, wenn es relevant ist sonst wird es verworfen. Alle relevanten Triple werden in einer separaten  $n$ -triple Datei gespeichert und später in ein SailRepository (Datenbanktyp für RDF4J). Zuletzt werden alle Datenbankdateien für die Auslieferung komprimiert. Jedes offline Datenpaket erhält eine Revisions-ID (Hash des Inhalts) um es zu identifizieren. Das Werkzeug schreibt auch ein JSON-Objekt um es mit dem Offline-Data-Synchronizer zu nutzen.

RDF4J ist ein open-source Java Framework für Linked Data, das eine aktive Community hat und regelmäßige Updates erhält. Somit war es die beste Option auch wenn es keine Hinweise gab, ob es effektiv auf einem Android-System arbeitet. Es wurde RDF4J Multilingual Question Answering mit Limited Resources v1.0.3 genutzt welches

<sup>24</sup>Demo App: [https://drive.google.com/open?id=1u09qDCEQv7dNv6\\_0ERFm5hiFSxzsBzRa](https://drive.google.com/open?id=1u09qDCEQv7dNv6_0ERFm5hiFSxzsBzRa)

ein Rückport der aktuellen RDF4J Version für ältere Java1.7 Umgebungen ist und Android unterstützt. OQA nutzt zwei Synchronizer zum Speichern der Linked Data auf dem mobilen Endgerät. Einen serverseitigen Synchronizer und einen mobilseitigen Synchronizer.

Erscheinungsbild: Die App hat eine sehr einfache Benutzeroberfläche. Wenn ein Nutzer die App startet, lädt die App mit Hilfe des Offline Data Managers für Offline QA die Datenbank herunter. Anschließend fragt sie den Nutzer nach der Berechtigung, Zugang zu Kontakten und Kalender des Nutzers zu erhalten. Das Hauptmenü hat drei Bereiche:

1. QA-Bereich: Hier kann der Nutzer im Offline- oder Online-Modus Fragen stellen mit Hilfe der Tastatur oder der Spracherkennung.
2. Kontakte-Bereich: Hier findet der Nutzer alle seine Kontakte und Anrufe tätigen
3. Kalender-Bereich: Hier kann der Nutzer seine Termine und Ereignisse prüfen

### Auswertung

*Batteriekapazität:* Erstens werden Ergebnisse des Batterieverbrauchs berücksichtigt. Mobile Endgeräte haben eine eingeschränkte Batteriekapazität. Nutzer möchten keine App verwenden, die viel Energie verbraucht. Um den Energieverbrauch zu messen, wurde Battery Historian<sup>25</sup> verwendet. Im Online-Modus werden ein Online-SPARQL-Endpunkt als Datenbank und ein online-Webservice zum Analysieren der Fragen genutzt, was beides keinen messbaren Energieverbrauch ergibt. Im Offline-Modus verbraucht OQA nur 5,28% Batteriekapazität um 200 Fragen zu beantworten. Zweitens wurde OQA mit LCQuAD und anderen deutschen Test-Datensätzen überprüft.

*Performance:* Wir haben gegen LCQuAD getestet, aber die Ergebnisse waren nicht eindeutig. Deshalb wurde der Algorithmus mit weiteren englischen Fragen getestet. Wenn man dem exakten Wortlaut der Label der Entitäten und der Properties folgt, kann man brauchbare Ergebnisse erzielen. Weicht man aber von dieser Vorgehensweise ab, sind die Antworten schnell falsch. Hinzu kommt, dass einige Daten im Online-Datensatz nicht einmal verfügbar sind. Der Algorithmus erkennt dies jedoch nicht, so dass etwas völlig Absurdes ausgegeben wird. Wenn die Label aus mehreren Wörtern bestehen, ist es wahrscheinlicher, dass die korrekte Entität gefunden wird. Der Algorithmus ist zudem stark abhängig vom genutzten Fragewort. Wenn die Antwort einen Datentyp hat, der nicht berücksichtigt wurde, kann der Algorithmus sie nicht finden. Aus diesem Grund wurde mit einem weiteren Algorithmus experimentiert, der nach allen Properties sucht und dann diejenigen bevorzugt, die den korrekten Datentyp aufweisen, anstatt sich auf diesen zu beschränken. Auf Grund der höheren Anzahl von Properties die geprüft werden müssen, führte dies zu einer beträchtlich schlechteren Laufzeit. Darum wurde der Ansatz wieder verworfen. Der Algorithmus konnte verbessert werden, in dem nicht nur auf die Entität und seine Properties geschaut wurde, sondern auch auf die Nachbarknoten im Graph und dann daraus Schlußfolgerungen gezogen wurden. Diese umfangreiche Überprüfung

---

<sup>25</sup>Battery Historian <https://developer.android.com/studio/profile/battery-historian>

wäre sehr viel rechenintensiver und darum für den Offline-Modus ungeeignet. Im Online-Modus kann es jedoch in Betracht gezogen werden. Die Ergebnisse dort sind daher auch viel akkurater.

### *Performance des Offline-Fragebeantwortungssystem mit dem Köln-Datensatz*

Die App ist in der Lage 35 von 46 Fragen zu beantworten. Das liegt daran, dass das Offline-Fragebeantwortungssystem sich auf einfache Fragen mit binären Relationen beschränkt, auch bekannt als Triple. Das System kann die Frage korrekt beantworten, wenn die gleichen Label für Entitäten und Properties genutzt werden. Die Ergebnisse können hier gefunden werden: <https://github.com/dice-group/qamel/tree/master/app>

**Vergleich mit angestrebten Zielen** Die Universität Leipzig war an allen Arbeitspaketen beteiligt. Ziel von AP1 war eine Machbarkeitsstudie, eine Architektur sowie die Erarbeitung von Evaluationskriterien für QAMEL. Die Ergebnisse dieses Arbeitspaketes sind in Abschnitt 2.1.2 detailliert. In AP2 wurde eine Sammlung von Werkzeugen zur multilingualen Extraktion - deutsch, englisch, koreanisch - Konvertierung von Tabellen und kurzen Texten (Fragen) erzeugt sowie ein Datenauswahlmodul zur Verkleinerung von Modellen, welche auf mobilen Endgeräten laufen sollen. AP3 fokussierte sich auf die Erarbeitung mehrerer Frageformulierungshilfen. Dazu wurde eine Menge an Systemen implementiert. In Arbeitspaket 4 wurden zwei Question-Answering-Systeme zur Beantwortung natürlichsprachlicher Nutzeranfragen entworfen, implementiert und evaluiert. Zusätzlich wurde das GERBIL Projekt zur Evaluation von Fragen erweitert. In Arbeitspaket 5 erfolgte ein Vergleich möglicher Triple Stores für mobile Endgeräte sowie die Adaption eines ausgewählten. Abschließend wurde in AP6 ein Prototyp implementiert und evaluiert, so dass das Zusammenspiel aller Komponenten evaluiert werden konnte. Die Universität Leipzig unterstützte den Konsortialführer sowie alle Partner beim Projektmanagement durchgehend über den Projektzeitraum (insbesondere in den APs 7 bis 9). Es wurden alle im Antrag genannten Ziele erreicht.

## 2.2 Zahlenmäßiger Nachweis

Von Seiten der Universität Leipzig wurde die mit dem Projektträger abgeprochene, geänderte Zeitplanung eingehalten. Die Gesamtprojektkosten betragen 296.079,17 Euro, d.h. 1.317,05 Euro mehr als geplant.

## 2.3 Notwendigkeit und Angemessenheit der geleisteten Arbeit

Die von der Universität Leipzig geleisteten Arbeiten erfolgten entsprechend der im Laufe von QAMEL spezifizierten Anforderungen <sup>26</sup>. Im Folgenden werden die Hauptanforderungen zusammengefasst. Es wird zusätzlich aufgezeigt, wie QAMEL Technologien diese

---

<sup>26</sup>D1.3 Anforderungsanalyse: <https://drive.google.com/file/d/0BzkLD3pa7T5XZko0QkFVZVpKZ3M/view>

erfüllen.

**AP2 - Informationsextraktion** Für QAMEL war es notwendig ein Informationsextraktionssystem zu schaffen, welches

1. Wissensextraktion aus unstrukturierten Quellen (Texten) nach RDF leisten kann;
2. Entitäten zu beliebigen Wissensbasen verlinken kann;
3. Sensordaten von mobilen Endgeräten extrahieren kann, um Fragen in Bezug auf Datum, Uhrzeit, Ort beantworten zu können;
4. Informationsextraktion aus unstrukturierten Textquellen leisten kann, wo insbesondere Produktnamen, Gerätemarken, Modelle und Versionen erkannt werden;
5. Mehrsprachiges Named Entity Recognition kann, insbesondere in Englisch und Deutsch;
6. Informationsextraktion aus semi-strukturierten Daten unterstützt, d.h. die Transformation von Informationen aus Tabellen nach RDF unterstützt.

Die Anforderung 1 bis 6 ergaben sich aus der Notwendigkeit der Darstellung der Daten als RDF. Zu Beginn des Projektes stellte keins der existierenden nicht-kommerziellen Lösung für die Extraktion von RDF aus unstrukturierten bzw. semi-strukturierten Daten eine befriedigende Lösung zur Verfügung. Insbesondere waren Multilingualität, die Erfassung von Sensordaten, die Benutzung unterschiedlicher Wissensbasen und die Extraktion aus Tabellen nicht unterstützte Feature. Zusätzlich war die Genauigkeit sowie die Vollständigkeit der Werkzeuge nicht zufriedenstellend. Für die Anforderungen 1 und 4 wurde FOX für kurze Texte, bspw. Fragesätze erweitert und debuggt. Performanzwerte wurden von FOX durch die intelligente Zusammenführung der Ergebnisse existierender Werkzeuge erreicht. Somit wurde nicht versucht, das “Rad neu zu erfinden”, sondern es wurde auf existierende Technologien aufgebaut. Die Ergebnisse der Evaluation von FOX bestätigen, dass die intelligente Zusammenführung existierender Werkzeuge die geforderten Ziele bezüglich Genauigkeit und Vollständigkeit erreicht. Für die Anforderungen 2 und 5 wurde AGDISTIS zu MAG erweitert und ebenfalls debuggt. Anforderung 3 wurde mittels einer QAMEL-Eigenentwicklung erfüllt, während Anforderung 6 durch die Erweiterung des TAIPAN Frameworks sichergestellt wurde. Alle Techniken sind in Abschnitt 2.1 näher beschrieben.

**AP3 - Frageformulierung** Für eine QAMEL-taugliche Frageformulierung wurden Anforderungen erhoben, welche folgende Möglichkeiten bieten sollten:

1. Die unterstützten Eingabesprachen sind Koreanisch und Englisch.
2. Falsch geschriebene Eingaben zu korrigieren, durch Training auf einer großen Menge an Textdaten.

3. Erstellung einer Liste von möglichen Abfragen, die der Benutzerabsicht bei der ersten Eingabe entsprechen können.
4. Nutzung der Grundfunktionen der Spracherkennung für Anfragen.

Anforderung 1 ergab sich aus der Komposition des Konsortiums und wurde erfolgreich in den drei Anwendungsszenarien umgesetzt. Insbesondere sichert KAIST die Korrektheit koreanischer Eingaben ab. Anforderung 2 ergab sich aus den Erfahrungen der industriellen Partner, welche durch Korrektur der Nutzer bessere Suchergebnisse liefern wollten. Dies funktioniert nur bei textueller Eingabe. Eine entsprechende Programm-bibliothek wurde von ULEI für QAMEL implementiert. Ebenso wurde Anforderung 3 abgearbeitet. Eine Autovervollständigungsfunktion wurde von ULEI für QAMEL implementiert. Die Nutzung von Spracherkennung (Anforderung 4) wurde nach Evaluation der zur Verfügung stehenden industriellen Anforderungen und akademischen Erkenntnisse über eine QAMEL-eigene Anbindung der Android SDK realisiert.

**AP4 - Fragebeantwortungssysteme für den Online- und Offline-Betrieb** Für QAMEL war es notwendig, ein QA System zu entwickeln, dass

1. Natürlichsprachliche Anfragen beantwortet,
2. Eine hohe Genauigkeit auf Testdaten aufweist,
3. Offline-fähig ist,
4. Antworten für Benutzer zur Verfügung stellt, so dass diese mehr als eine URI sehen,
5. Der Kontext bei der Formulierung von Fragen an das System (also Sensorinformationen) berücksichtigt werden.

Durch das unterschiedliche technische Hintergrundwissen der Nutzer ergeben sich diverse Anforderungen an das natürlichsprachliche Interface für Abfragen in Form von Fragesätzen (Anforderung 1, inhärent). Anforderung 2 ergibt sich konsequenterweise, da ein Suchsystem im Idealfall nur korrekte Antworten geben soll. Die Offline-Fähigkeit (Anforderung 3) stellt einen der zentralen Forschungsschwerpunkte des Projektes dar. Gerade in so genannten Hands-free Szenarien, wo Spracheingaben über Fragesätzen gebraucht werden also bspw. beim Autofahren, in Werkhallen etc., ist eine Internetkonnektivität nicht gegeben. Darum müssen die Basisfunktionen eines Suchsystems auch offline funktionieren, um einen Nutzen zu garantieren. In den oben genannten Situationen ist es insbesondere wichtig einen Kontext der Antwort noch einmal mitzugeben, bspw. Bilder, Beschreibungstext u.ä., da hier mit minimaler Interaktion und kognitiver Anstrengung das Ergebnis mitgeteilt werden muss. Anforderung 5 bezieht sich insbesondere auf Anwendungen auf mobilen Endgeräten bei denen der Nutzer auf seine Situation bezogene Fragen stellt, bspw: *Wie ist die Telefonnummer meiner Mutter?* oder *Wie heißt das Gebäude vor mir?* Die Anforderungen wurden durch unsere QA Systeme erfüllt. Insbesondere erfüllt der Machbarkeitsprototyp OQA die Anforderungen 1,3,4,5 und TeBaQA die Anforderungen 1,2,4.

**AP5 - Auswertung von verfügbaren Triple-Stores zum Betrieb auf mobilen Endgeräten** Durch das Ausscheiden von Projektpartnern beschränkte sich AP 5 auf eine Analyse vorhandener Systeme, bei der die Hauptanforderung war:

1. Laufzeit-Effizienz: Das Suchsystem für RDF-Daten sollte in der Lage sein, Antworten innerhalb von 3 Sekunden zurückzugeben

Die in AP5 dann vorgenommene Analyse und die Anpassung von RDF4J an das Android Betriebssystem erfüllen diese Anforderung.

**AP6 - Anwendungsstudie Enzyklopädische Wissensbasis** Für QAMEL war es notwendig einen Anwendungsfall zu implementieren, welcher

1. den Einsatz von natürlichsprachlichen Suchsystemen unterstützt,
2. die von ULEI und den Partnern entwickelten Komponenten integriert und
3. über eine Nutzerstudie evaluiert wird.

Die durch ULEI in AP6 entwickelte mobile App<sup>27</sup>, welche auf Android basiert, erfüllt alle diese Anforderungen. Insbesondere basiert die App auf den in 1) AP4 implementierten OQA-Algorithmus, 2) einen eigenen mobilen Linked Data-Triple-Store (AP5), 3) einen einfachen, aber dennoch effektiven mehrsprachigen Algorithmus für die Transformation von natürlicher Sprache in SPARQL, welcher offline funktioniert (AP4/AP3). OQA fokussiert sich auf eine geringe Ressourcenlast, d.h. Batterie- und Speicherverbrauch. Benutzern wird es dabei ermöglicht 4) eine Teilmenge der DBpedia abzufragen (AP2) sowie weitere auf dem mobilen Endgerät zur Verfügung stehende Daten (AP2).

## 2.4 Voraussichtlicher Nutzen

Alle von ULEI im Rahmen von QAMEL entwickelten Technologien stehen der Öffentlichkeit als Open-Source oder Open-Service Frameworks zur Verfügung, siehe <https://github.com/dice-group/qamel>. Wir planen u.a. eine erweiterte Version von Wikipedia zu veröffentlichen, welche mit den in QAMEL entwickelten Informationsextraktoren erstellt wurde. So soll der Nutzen und die Nutzbarkeit unserer Technologien noch klarer demonstriert werden. Zusätzlich zum gesamten QAMEL Framework können die einzelnen Komponenten von QAMEL unabhängig von einander genutzt werden.

Der Hauptnutzen von FOX besteht in der Extraktion von strukturierten Daten aus Texten und Fragen. Das FOX Projekt wurde bereits erfolgreich in mehrere Lösungen eingebettet. Zum Beispiel nutzt GeoLift<sup>28</sup> die Ergebnisse aus FOX um geographische Daten aus langen RDF-Literalen zu extrahieren. Ein Microsoft SharePoint Plugin für das Tagging von eMails, Terminen, Dokumenten wurde von Studenten der Universität Leipzig entwickelt.

---

<sup>27</sup><https://github.com/dice-group/qamel/tree/master/app>

<sup>28</sup><http://github.com/AKSW/GeoLift>

TAIPAN ist ein neuartigen Ansatz zur Erstellung von RDF aus Web-Tabellen. So können Menschen die im Internet enthaltenen Tabellen, bspw. über einen WebCrawl, semantisch aufbereiten und im Rahmen von strukturierten, maschinellen Lernmethoden einsetzen. TAIPAN kann über eine Docker oder REST Schnittstelle eingebunden werden und so in viele wissenschaftliche und industrielle Anwendungen mit minimalem Aufwand integriert werden.

Die Fragebeantwortungssysteme OQA und TeBaQA haben verschiedene Nutzen. Während OQA der Prototyp eines mobilen, offline-fähigen QA Systems ist und damit insbesondere in Situationen eingesetzt werden kann, wo dauerhaft keine großen Rechenressourcen oder Internetanbindungen zur Verfügung stehen. Die prototypische Android-App kann einfach und schnell installiert und somit evaluiert werden. TeBaQA hingegen benötigt größere Rechenressourcen, lernt jedoch aus dem zur Verfügung stehenden Datenmaterial. Das heißt, je mehr Trainingsdaten es erhält, desto höher ist seine Performance. Damit kann TeBaQA gerade in Datenintensiven-Bereichen (B2C) eingesetzt werden. Eine Anpassung des Source Codes ist nicht notwendig, da TeBaQA allein mittels neuer Daten an eine Domäne angepasst werden kann. TeBaQA wird bereits in weitere Forschungsprojekten weiterentwickelt und angepasst.

GERBIL QA ist ein Benchmarking-System für Fragebeantwortungssysteme. Das System bildet mittlerweile das Backbone für mehrere, internationale QA Wettbewerbe und erleichtert Wissenschaftlern die Evaluation ihres QA Systems zusehens.

Kundenspezifisch angepasste Versionen der QAMEL Module bzw. in QAMEL weiterentwickelten Systeme und Datensätze werden von den Anwendungspartnern in der Zukunft versucht zur Verwertung zu benutzen. Natürlich dienen alle in QAMEL geleisteten Arbeiten auch der wissenschaftlichen Verwertung und Anschlussfähigkeit.

### 2.5 Fortschritt bei anderen Stellen

Im Rahmen der QAMEL Veröffentlichungen (siehe Abschnitt 2.6) wurde der Fortschritt bei anderen Stellen eingehend untersucht. Im Folgenden wird ein Überblick über diesen Fortschritt gegeben. Detaillierte Angaben können den zitierten Veröffentlichungen entnommen werden.

#### 2.5.1 Wissensextraktion

In den letzten Jahren wurde eine Vielzahl von Wissensextraktions-Werkzeugen entwickelt. Frameworks wie DBpedia Spotlight<sup>29</sup> und Wikipedia Miner<sup>30</sup> zielen darauf ab, alle Erwähnung von Wikipedia Entitäten zu erkennen. Sie sind im Gegenteil zu FOX jedoch nicht in der Lage Ressourcen bzw. Relationen zu erkennen, welche insbesondere nicht in Wikipedia vorkommen. Somit können sie im industriellen Sektor nur mit Anpassungen genutzt werden. Werkzeuge wie der Stanford Named Entity Recognizer<sup>31</sup> und der

---

<sup>29</sup><http://spotlight.dbpedia.org>

<sup>30</sup><http://wikipedia-miner.cms.waikato.ac.nz/>

<sup>31</sup><http://nlp.stanford.edu/software/CRF-NER.shtml>

Illinois Named Entity Tagger<sup>32</sup> werden ständig weiterentwickelt und erzielen immer bessere Precision und Recall-Werte [52]. Da sie in FOX eingebettet sind, bedeutet dies, dass FOX fortwährend besser wird. Kommerzielle Lösungen wie Alchemy<sup>33</sup> and Extractiv<sup>34</sup> können genauso in FOX eingebettet werden. Details können [29] entnommen werden.

### 2.5.2 Frageformulierung

Bibliotheken zur Korrektur, Spracherkennung oder Autovervollständigung werden in vielen kommerziell genutzten und frei verfügbaren Produkten eingesetzt. Durch Fortschritte in der Spracherkennung der Android SDK wird durch deren Anbindung auch QAMEL verbessert.

### 2.5.3 Fragebeantwortung

QA über Wissensgraphen wird immer noch üblicherweise als ein Semantic Parsing-Problem behandelt, bei dem das System eine formale Anfrage zurückgibt, z.B. SPARQL, die auf der KG ausgeführt werden kann, um die gewünschte Antwort abzurufen [9, 55, 40, 47]. Dubey et al. [9] verwenden eine Reihe von kanonischen Mustern als interne Frage-Darstellung, die in eine ausführbare SPARQL-Abfrage umgewandelt werden können. Cui et al. [5] nutzen zahlreiche prädikatspezifische, aber entitätsunabhängige Fragemustern, um einfache faktenverdächtige Fragen zu beantworten. Eine alternative Familie von Ansätzen versucht die Frage direkt in eine logische Form, die über einem Wissensgraphen ausführbar ist zu übersetzen. Zum Beispiel konstruiert der Neural Programmer [17] eine Liste von Kandidaten (bestehend aus Operationen wie count, select, argmax, etc.), die direkt über die KG ausführbar sind, unter Verwendung eines Sequence-to-Sequence-Netzwerks, das mit einem Key-Value-Store [39] erweitert wurde, um die Kompositionalität der Abfragesprache zu behandeln. Mehrere Varianten dieser Architekturen [56, 27, 1] wurden vorgeschlagen. Allerdings erreichen die meisten Systeme immer noch keinen F-measure über 0.43 [43].

### 2.5.4 Mobile RDF Triple Stores

Im Bereich mobiler RDF bzw. Triple Stores, also Datenbanken zur Speicherung von RDF auf mobilen Endgeräten mit begrenzten Ressourcen (insbesondere Festplattengeschwindigkeit, RAM und CPU) konnten keinerlei Fortschritte erkannt werden. Ein Hardwareentwicklungstrend in Richtung leistungsfähigerer Hardware in dem Ausmaß, wie er in QAMEL gebraucht werden würde, ist nicht abzusehen. Die wissenschaftlich/technische Ausrichtung bewegt sich eher in Richtung Datenselektion bzw. -komprimierung, um Daten- bzw. KI-modelle auf mobile Endgeräte zu bringen. Beide Forschungsrichtungen sind jedoch noch sehr weit am Anfang und meist proprietär.<sup>35</sup>

---

<sup>32</sup>[http://cogcomp.cs.illinois.edu/page/software\\_view/4](http://cogcomp.cs.illinois.edu/page/software_view/4)

<sup>33</sup><http://www.alchemyapi.com/api/>

<sup>34</sup><http://extractiv.com/>

<sup>35</sup><http://time.com/5585259/google-io-2019/>, abgerufen am 15. Mai 2019

## 2.6 Erfolgte und geplante Veröffentlichungen

### 2.6.1 Erfolgte Veröffentlichungen

Im Rahmen von QAMEL entstanden die unten folgenden Veröffentlichungen. Diese Veröffentlichungen gewannen unter anderem den "Runner Up Award" bei der KESW 2016 und decken alle Arbeitspakete sowie deren Erkenntnisse und Evaluationen ab. Eine entsprechende Zuordnung wurde in den Deliverables des Projektes bereits vorgenommen: <http://qamel.eu>.

1. Ricardo Usbeck, Michael Röder, Michael Hoffmann, Felix Conrads, Jonathan Huthmann, Axel-Cyrille Ngonga-Ngomo, Christian Demmler, and Christina Unger. Benchmarking question answering systems. *Semantic Web Journal*, 2018
2. Ricardo Usbeck, Michael Röder, Peter Haase, Artem Kozlov, Muhammad Saleem, and Axel-Cyrille Ngonga Ngomo. Requirements to modern semantic search engines. In *KESW*, 2016
3. Ricardo Usbeck and Axel-Cyrille Ngonga Ngomo. Qamel – question answering on mobile devices. In *ESWC, EU networking session*, 2016
4. Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Jin-Dong Kim, Key-Sun Choi, Philipp Cimiano, Irimi Fundulaki, and Anastasia Krithara, editors. *Joint Proceedings of BLINK2017: 2nd International Workshop on Benchmarking Linked Data and NLIWoD3: Natural Language Interfaces for the Web of Data co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 21st - to - 22nd, 2017*, volume 1932 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2017
5. Ricardo Usbeck, Erik Körner, and Axel-Cyrille Ngonga Ngomo. Answering boolean hybrid questions with hawk. In *NLIWOD workshop at International Semantic Web Conference (ISWC), including erratum and changes*, 2015
6. Ricardo Usbeck, Michael Hoffmann, Michael Röder, Jens Lehmann, and Axel-Cyrille Ngonga Ngomo. Using multi-label classification for improved question answering. *CoRR*, abs/1710.08634, 2017
7. Ricardo Usbeck, Jonathan Huthmann and Nico Duldhardt, and Axel-Cyrille Ngonga Ngomo. Self-wiring question answering systems. *CoRR*, 1611.01802, 2016
8. Muhammad Saleem, Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngonga Ngomo. Sparql querying benchmarks. In *Tutorial at ISWC*, 2016
9. Muhammad Saleem, Samaneh Nazari Dastjerdi, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. Question answering over linked data: What is difficult to answer? what affects the f scores? In *Natural Language Interfaces workshop at ISWC*, 2017

10. Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. Gerbil - benchmarking named entity recognition and linking consistently. *Semantic Web*, 9(5):605–625, 2018
11. Giulio Napolitano, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. The scalable question answering over linked data (sqa) challenge 2018. In *Semantic Web Challenges - 5th SemWebEval Challenge at ESWC 2018, Heraklion, Greece, June 3-7, 2018, Revised Selected Papers*, pages 69–75, 2018
12. Axel-Cyrille Ngonga Ngomo, Michael Hoffmann, Ricardo Usbeck, and Kunal Jha. Holistic and scalable ranking of rdf data. In *2017 IEEE International Conference on Big Data*, page 10, 2017
13. Diego Moussallem, Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngonga Ngomo. Mag: A multilingual, knowledge-base agnostic and deterministic entity linking approach. In *K-CAP 2017: Knowledge Capture Conference*, page 8, 2017
14. Diego Moussallem, Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngonga Ngomo. Entity linking in 40 languages using mag. In *The Semantic Web, ESWC 2018, Lecture Notes in Computer Science*, 2018
15. Jin-Dong Kim, Christina Unger, Axel-Cyrille Ngonga Ngomo, André Freitas, Younggyun Hahm, Jiseong Kim, Sangha Nam, Gyu-Hyun Choi, Jeong uk Kim, Ricardo Usbeck, et al. Okbqa framework for collaboration on developing natural language question answering systems. 2017
16. Jin-Dong Kim, Christina Unger, Axel-Cyrille Ngonga Ngomo, André Freitas, Younggyun Hahm, Jiseong Kim, Gyu-Hyun Choi, Jeonguk Kim, Ricardo Usbeck, Myoung-Gu Kang, and Key-Sun Choi. Okbqa: an open collaboration framework for development of natural language question-answering over knowledge bases. In *Proceedings of the ISWC 2017 Posters & Demonstrations and Industry Tracks co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 23rd - to - 25th, 2017.*, 2017
17. Konrad Höffner, Sebastian Walter, Edgard Marx, Ricardo Usbeck, Jens Lehmann, and Axel-Cyrille Ngonga Ngomo. Survey on challenges of question answering in the semantic web. *Semantic Web Journal*, 8(6), 2017
18. Timofey Ermilov, Diego Moussallem, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. Genesis – a generic rdf data access interface. In *WI '17 - IEEE/WIC/ACM International Conference on Web Intelligence*, page 7, 2017
19. Dennis Diefenbach, Ricardo Usbeck, Kamal Deep Singh, and Pierre Maret. Scalable approach for computing semantic relatedness using semantic web data. In *Proceedings of the International Conference on Web Intelligence, Mining and Semantics, WIMS '16*, 2016

20. Key-Sun Choi, Luis Espinosa Anke, Thierry Declerck, Dagmar Gromann, Jin-Dong Kim, Axel-Cyrille Ngonga Ngomo, Muhammad Saleem, and Ricardo Usbeck, editors. *Joint proceedings of the 4th Workshop on Semantic Deep Learning (SemDeep-4) and NLIWoD4: Natural Language Interfaces for the Web of Data (NLIWOD-4) and 9th Question Answering over Linked Data challenge (QALD-9) co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, California, United States of America, October 8th - 9th, 2018*, volume 2241 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2018
21. Ricardo Usbeck, Ria Hari Gusmita, Axel-Cyrille Ngonga Ngomo, and Muhammad Saleem. 9th challenge on question answering over linked data (qald-9) (invited paper). In *Joint proceedings of the 4th Workshop on Semantic Deep Learning (SemDeep-4) and NLIWoD4: Natural Language Interfaces for the Web of Data (NLIWOD-4) and 9th Question Answering over Linked Data challenge (QALD-9) co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, California, United States of America, October 8th - 9th, 2018.*, pages 58–64, 2018
22. Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Felix Conrads, Michael Röder, and Giulio Napolitano. 8th challenge on question answering over linked data (qald-8). In *Joint proceedings of the 4th Workshop on Semantic Deep Learning (SemDeep-4) and NLIWoD4: Natural Language Interfaces for the Web of Data (NLIWOD-4) and 9th Question Answering over Linked Data challenge (QALD-9) co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, California, United States of America, October 8th - 9th, 2018.*, pages 51–57, 2018

### 2.6.2 Geplante Veröffentlichungen

Die QAMEL-Ergebnisse werden aktiv weitergepflegt und stetig erweitert. In dieser Hinsicht geplant ist weitere Publikationen zu GERBIL QA, OQA und TeBaQA zu veröffentlichen und die Benchmark-Ergebnisse und Datensätze als Ressourcenpublikationen voranzutreiben.

## Literaturverzeichnis

1. Mohnish Dubey, Sourish Dasgupta, Ankit Sharma, Konrad Höffner, and Jens Lehmann. Asknow: A framework for natural language query formalization in sparql. In *International Semantic Web Conference*, pages 300–316. Springer, 2016
2. Kun Xu, Sheng Zhang, Yansong Feng, and Dongyan Zhao. Answering natural language questions via phrasal semantic parsing. In *Natural Language Processing and Chinese Computing*, pages 333–344. Springer, 2014
3. Christina Unger, Lorenz Bühmann, Jens Lehmann, Axel-Cyrille Ngonga Ngomo, Daniel Gerber, and Philipp Cimiano. Template-based question answering over RDF data. In *Proceedings of the 21st World Wide Web Conference 2012, WWW 2012, Lyon, France, April 16-20, 2012*, pages 639–648, 2012
4. Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Lorenz Bühmann, and Christina Unger. HAWK - hybrid question answering using linked data. In *The Semantic Web. Latest Advances and New Domains - 12th European Semantic Web Conference, ESWC 2015, Portoroz, Slovenia, May 31 - June 4, 2015. Proceedings*, pages 353–368, 2015
5. Wanyun Cui, Yanghua Xiao, Haixun Wang, Yangqiu Song, Seung-won Hwang, and Wei Wang. Kbqa: learning question answering over qa corpora and knowledge bases. *Proceedings of the VLDB Endowment*, 10(5):565–576, 2017
6. Chen Liang, Jonathan Berant, Quoc Le, Kenneth D. Forbus, and Ni Lao. Neural symbolic machines: Learning semantic parsers on freebase with weak supervision. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23–33. Association for Computational Linguistics, 2017
7. Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448, 2015
8. Ricardo Usbeck, Ria Hari Gusmita, Axel-Cyrille Ngonga Ngomo, and Muhammad Saleem. 9th challenge on question answering over linked data (qald-9) (invited paper). In *Joint proceedings of the 4th Workshop on Semantic Deep Learning (SemDeep-4) and NLIWoD4: Natural Language Interfaces for the Web of Data (NLIWOD-4) and 9th Question Answering over Linked Data challenge (QALD-9) co-located with 17th International Semantic Web Conference (ISWC 2018), Monterey, California, United States of America, October 8th - 9th, 2018.*, pages 58–64, 2018
9. Timofey Ermilov, Diego Moussallem, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. Genesis – a generic rdf data access interface. In *WI '17 - IEEE/WIC/ACM International Conference on Web Intelligence*, page 7, 2017

10. Diego Moussallem, Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngonga Ngomo. Entity linking in 40 languages using mag. In *The Semantic Web, ESWC 2018, Lecture Notes in Computer Science*, 2018
11. Qingqing Cao, Noah Weber, Niranjan Balasubramanian, and Aruna Balasubramanian. Deqa: On-device question answering. 2019
12. Dennis Diefenbach, Kamal Deep Singh, and Pierre Maret. WDAqua-core0: A Question Answering Component for the Research Community. In *Semantic Web Challenges*, pages 84–89, 2017
13. Konrad Höffner, Sebastian Walter, Edgard Marx, Ricardo Usbeck, Jens Lehmann, and Axel-Cyrille Ngonga Ngomo. Survey on challenges of question answering in the semantic web. *Semantic Web Journal*, 8(6), 2017
14. Daniil Sorokin and Iryna Gurevych. End-to-end representation learning for question answering with weak supervision. In *Semantic Web Challenges*, pages 70–83, 2017
15. Ricardo Usbeck, Ria Hari Gusmita, Axel-Cyrille Ngonga Ngomo, and Muhammad Saleem. 9th Challenge on Question Answering over Linked Data (QALD-9). In *Joint proceedings of SemDeep-4 and NLIWOD-4*, pages 58–64, 2018
16. Manaal Faruqui and Sebastian Padó. Training and evaluating a german named entity recognizer with semantic generalization. In *Proceedings of KONVENS 2010*, Saarbrücken, Germany, 2010
17. Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370. Association for Computational Linguistics, 2005
18. Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014
19. L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *CoNLL*, 6 2009
20. Jason Baldridge. The opennlp project. URL: <http://opennlp.apache.org/index.html>, (accessed 17 May 2017), 2005
21. David Nadeau. Balie—baseline information extraction: Multilingual information extraction from text with machine learning and natural language techniques. Technical report, Technical report, University of Ottawa, 2005

22. Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*, 2013
23. Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Wencan Luo, and Lars Wesemann. Multilingual disambiguation of named entities using linked data. In *International Semantic Web Conference (ISWC), Demos & Posters*, 2014
24. Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Michael Röder, Sören Auer, Daniel Gerber, and Andreas Both. Agdistis - agnostic disambiguation of named entities using linked open data. In *European Conference on Artificial Intelligence*, page 2. 2014
25. Diego Moussallem, Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngonga Ngomo. Entity linking in 40 languages using mag. In *The Semantic Web, ESWC 2018, Lecture Notes in Computer Science*, 2018
26. Diego Moussallem, Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngonga Ngomo. MAG: A Multilingual, Knowledge-base Agnostic and Deterministic Entity Linking Approach. In *K-CAP 2017: Knowledge Capture Conference*, page 8. ACM, 2017
27. Daniel Gerber and Axel-Cyrille Ngonga Ngomo. Bootstrapping the linked data web. In *1st Workshop on Web Scale Knowledge Extraction @ ISWC 2011*, 2011
28. Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. Patty: A taxonomy of relational patterns with semantic types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1135–1145, Jeju Island, Korea, July 2012. Association for Computational Linguistics
29. René Speck and Axel-Cyrille Ngonga Ngomo. On extracting relations using distributional semantics and a tree generalization. In *Proceedings of The 21th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2018)*, 2018
30. Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014
31. Dan Roth, Wen-tau Yih, and Scott Wen-tau Yih. *Global Inference for Entity and Relation Identification via a Linear Programming Formulation*. MIT Press, November 2007
32. Axel-Cyrille Ngonga Ngomo, Norman Heino, Klaus Lyko, René Speck, and Martin Kaltenböck. Scms - semantifying content management systems. In *ISWC 2011*, 2011

33. René Speck and Axel-Cyrille Ngonga Ngomo. Ensemble learning for named entity recognition. In *The Semantic Web – ISWC 2014*, volume 8796 of *Lecture Notes in Computer Science*, pages 519–534. Springer International Publishing, 2014
34. René Speck and Axel-Cyrille Ngonga Ngomo. Ensemble Learning of Named Entity Recognition Algorithms using Multilayer Perceptron for the Multilingual Web of Data. In *K-CAP 2017: Knowledge Capture Conference*, page 4. ACM, 2017