The Eurostars Programme is powered by
EUREKA and the European Community

Schlussbericht zum Vorhaben "Keyword-Basierte Fragebeantwortung in Verteilten Dateninfrastrukturen"

im Rahmen des Eurostarsprojektes
E!9367 DIESEL
Gesamtprojekttitel:
"Distributed Search in Large Enterprise Data
Verteilte Suche in Unternehmensdaten"

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

Universität Leipzig, Augustusplatz 10
04109 Leipzig

Gefördert durch das
Bundesministerium für Bildung und Forschung (BMBF)
Förderkennzeichen: 01QE1512C
Projektlaufzeit: 01.09.2015 – 31.10.2018

29. April 2019

Inhaltsverzeichnis

1 Kurze Darstellung **4**
1.1 Aufgabenstellung 4
1.2 Voraussetzungen 4
1.3 Planung und Ablauf des Vorhabens 4
1.4 Wissenschaftlicher Stand, an den angeknüpft wurde 5
1.5 Zusammenarbeit mit anderen Stellen 5

2 Eingehende Darstellung **6**
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 21
2.3 Notwendigkeit und Angemessenheit der geleisteten Arbeit 21
2.4 Voraussichtlicher Nutzen 23
2.5 Fortschritt bei anderen Stellen 24
2.5.1 Stichwortsuche 24
2.5.2 Föderierte Anfragen 25
2.5.3 Wissensextraktion 25
2.6 Erfolgte und geplante Veröffentlichungen 26
2.6.1 Erfolgte Veröffentlichungen 26
2.6.2 Geplante Veröffentlichungen 29

Abbildungsverzeichnis

1 Architektur von DIESEL 7
2 Architektur von FOX 8
3 Architektur von Ocelot für RE 10
4 Vergleich der Query Ausführungszeiten auf FedBench. 14
5 Architektur von SESSA 16
6 Architektur von Autoindex 17
7 CostFed Ergebnisse im GENESIS Framework 20

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 `spouse`, `birthPlace`, `deathPlace` und `subsidiary` für die k besten Muster. 11

1 Kurze Darstellung

1.1 Aufgabenstellung

Im DIESEL Vorhaben wurde ein generisches und zeiteffizientes Framework für die Keyword-basierte Beantwortung von Fragen für strukturierte verteilte Datenquellen entwickelt. Insbesondere wurden folgende Kategorien von Methoden untersucht:

- Extraktionsalgorithmen zur Überführung von unstrukturierte Datenquellen in strukturierte Daten, d.h. RDF;[1]
- Verfahrung zur Auswahl von Datenquellen während der Verarbeitung von föderierten SPARQL Anfragen;[2]
- Evaluation von Verfahren zur Generierung von SPARQL[3] Anfragen aus Sequenzen von Schlüsselwortanfragen.[4]

Alle 18 Deliverables können auf der Projektwebseite `https://diesel-project.eu/` eingesehen werden, eine Demo ist unter `http://genesis-diesel.aksw.org/` verfügbar.

1.2 Voraussetzungen

Das Vorhaben wurde in 3 Jahren als Verbundprojekt durchgeführt. Die Leitung des Konsortiums übernahm die metaphacts GmbH. Weitere technische Projektpartner waren die Universität Leipzig (ULEI) und die zazuko GmbH aus der Schweiz. Die Projektlaufzeit wurde für die Universität Leipzig um 2 Monate kostenneutral verlängert.

1.3 Planung und Ablauf des Vorhabens

Das Vorhaben bestand aus sechs Arbeitspaketen (AP): Requirements Elicitation (AP1, M1 – M6), Federated Query Generation (AP2, M7 – M30), Duplicate-Aware Federated Query Processing (AP3, M6 – M30), Knowledge Extraction for Enterprise Data (AP4, M6 – M30), Use Cases (AP5, M31 – M36), Project Management (AP6, M1 – M36).

Die Verantwortlichkeit für AP1-6 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, Quetsal und CostFed sowie SESSA und Autoindex. Desweiteren erarbeitete die Universität Leipzig diverse Evaluationsdatensätze für AP2, welche in das GERBIL Framework für Extraktionsalgorithmen Eingang gefunden haben.[5] metaphacts erweiterte die metaphactory Platform mit eigenen Implementierungen sowie

[1] `https://github.com/dice-group/FOX`, `https://github.com/dice-group/AGDISTIS`
[2] `https://github.com/dice-group/CostFed`
[3] `http://www.w3.org/TR/rdf-sparql-query/`
[4] `https://github.com/dice-group/autoindex`, `https://github.com/dice-group/SESSA`
[5] `http://gerbil.aksw.org/gerbil/`

zur Integration der verschiedenen Tools der Partner und Zazuko steuerte als unbezahlter Partner Daten und Feedback zu den Fallstudien bei. Die Anwendungsfälle (engl. Use Cases) wurden von allen Partnern beigesteuert und unter Verwendung der implementierten Verfahren umgesetzt und evaluiert. 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 und Quetsal. Keins dieser Werkzeuge stellte die zu entwickelnden Funktionen zur Verfügung. Sie bildeten jedoch die technischen Grundlagen für die Implementierung von Teilen der entwickelten Verfahren. Neue Frameworks wie GERBIL, CostFed, SESSA und Autoindex entstanden während der Projektlaufzeit. Als Fachliteratur wurden die in den DIESEL Publikationen genannten Veröffentlichungen und Bücher verwendet. Als Informationsdienste dienten öffentliche Publikationsportale wie Google Scholar,[6] DBLP[7] und die ACM Digital Library[8], der IEEE Explorer[9] sowie die Bibliothek der Universität Leipzig.

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 SLUB Dresden, der TIB Hannover und des koreanischen KAIST organisiert. Desweiteren wurde sich mit Mitarbeitern der EU-Projekte HOBBIT 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 DIESEL.

[6] `http://scholar.google.com`
[7] `http://dblp.uni-trier.de/`
[8] `http://dl.acm.org/`
[9] `https://ieeexplore.ieee.org`

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 im Rahmen von Projekt- und Konferenzreisen entstandenen Sachausgaben verwendet. Es wurden ferner zwei Server erworben, auf dem die Entwicklung des gesamten Projekts stattfand. Auf diesen Servern werden auch alle Demonstratoren betrieben. Die Server werden auch nach der Projektlaufzeit ausschließlich für wissenschaftliche Zwecke genutzt.

2.1.2 Erzielte Ergebnisse

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

1. eine Architektur für Verteilte Suche in Unternehmensdaten,
2. eine Erweiterung eines Frameworks für die Extraktion von Entitäten und Relationen aus Volltexten,
3. ein effektives und effizientes System für föderierte SPARQL Anfragen,
4. eine effektive semantische Schlüsselwortsuchkomponente und
5. 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.

DIESEL Architektur Die DIESEL Architektur wurde aus realen Anforderungen aus Wirtschaft und Wissenschaft erarbeitet (AP1). Grundidee hinter dieser Architektur war die Bereitstellung eines einfach anpassbaren, generischen Frameworks, welches sich für spätere (z.B. kundenspezifische) Erweiterungen eignet. Das DIESEL-Framework hat eine mehrschichtige Architektur (siehe Abbildung 1). Die orangefarbenen Module entsprechen der tatsächlichen Entwicklung, die im Rahmen des Projekts durchgeführt wurde. Der Benutzer beginnt mit der Abfrage der Anwendung mittels natürlicher Sprache oder via Stichwortsuche in einer einzigen Oberfläche. Diese Schnittstelle ist auf jeden der DIESEL-Anwendungsfälle spezialisiert. Die Abfrage wird analysiert und in eine oder mehrere SPARQL-Abfragen innerhalb der semantischen Suchkomponente umgewandelt. Die SPARQL-Abfragen werden dann vom Föderierungssystem (z.B. Questsal oder Costfed) verarbeitet. Die föderierten Abfragen werden dann in den verschiedenen Endpunkten ausgeführt, die zuvor mit den in RDF transformierten Daten versorgt wurden. Das Datenextraktionsmodul besteht aus verschiedenen Technologien zur Extraktion und/oder

Transformation von Daten aus verschiedenen Informationsquellen wie Datenbanken, Dokumentrepositorien (csv, xml, pdf, etc.), etc. Einige dieser Quellen sind unstrukturierte Informationen und erfordern eine Extraktion von Entitäten aus Volltexten. Die Wahl der zu verwendenden Extraktionskomponenten hängt vom Anwendungsfall ab.

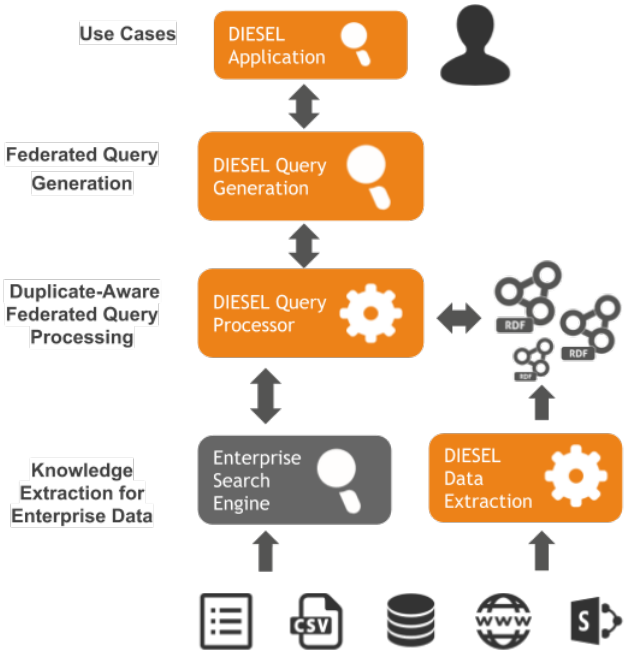

Abbildung 1: Architektur von DIESEL

Extraktion von Entitäten und Relationen aus Volltexten Die Extraktion von strukturierten Wissen aus unstrukturierten Datenquellen bildete den Kern von AP4 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 AP4 war die Entwicklung eines auf DBpedia basierenden Wissensextraktions Frameworks für lange Texte im Unternehmensbereich. 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 2) besteht aus drei Schichten:

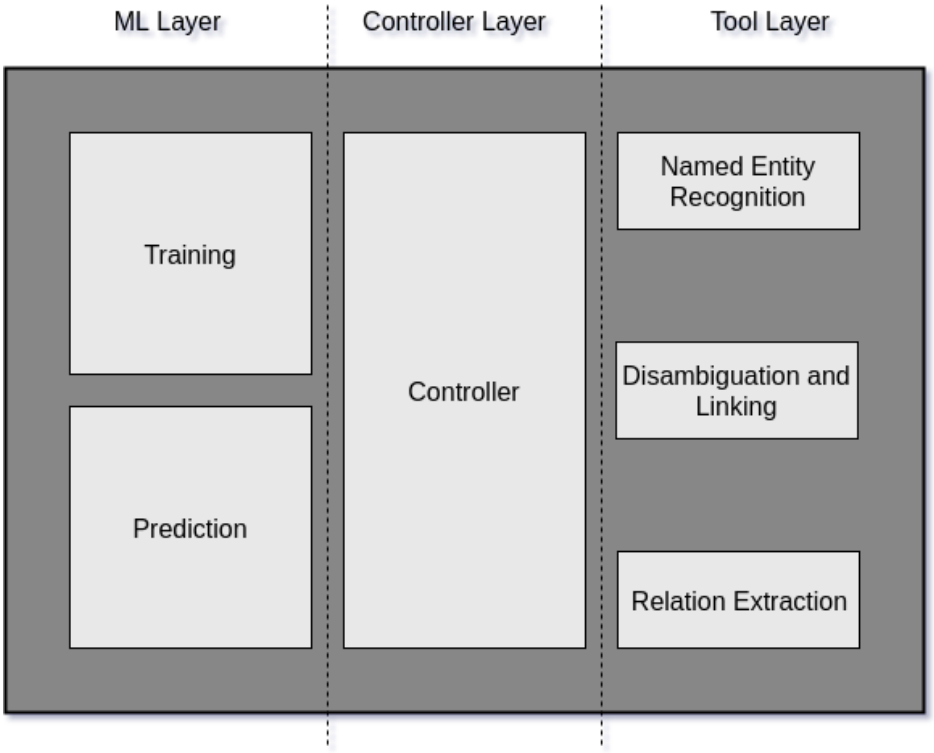

Abbildung 2: Architektur von FOX

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 unbekannten Zeichen). Die Eingabe wird dann an alle vom Nutzer ausgewählten Extraktionswerkzeuge weitergeleitet. Die Ergebnisse 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 [14, 15, 23][10], Illinois [37][11], OpenNLP [4][12],

[10] `http://nlp.stanford.edu:8080/ner/process`
[11] `http://cogcomp.cs.illinois.edu/page/demo_view/ner`
[12] `http://opennlp.apache.org/download.html`

Balie [31][13] und Spotlight [6][14].

b) Werkzeuge für NED: AGDISTIS [56, 57] und MAG [27, 28].

c) Werkzeuge für RE: Boa [17], Patty [32], Ocelot [49] und StanfordCoreNLP [24] mit [38].

Die in [35, 47, 48] vorgestellten Evaluationen 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 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	**92.73**	**95.23**	**68.81**	**87.55**	**90.99**	**90.70**	**93.09**	**63.36**	81.98	**90.28**
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	**83.74**	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.

Ein Überblick des Datenflusses von Ocelot ist in Abbildung 3 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.

[13] `http://balie.sourceforge.net/`
[14] `http://spotlight.sztaki.hu/downloads/`

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

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

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

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

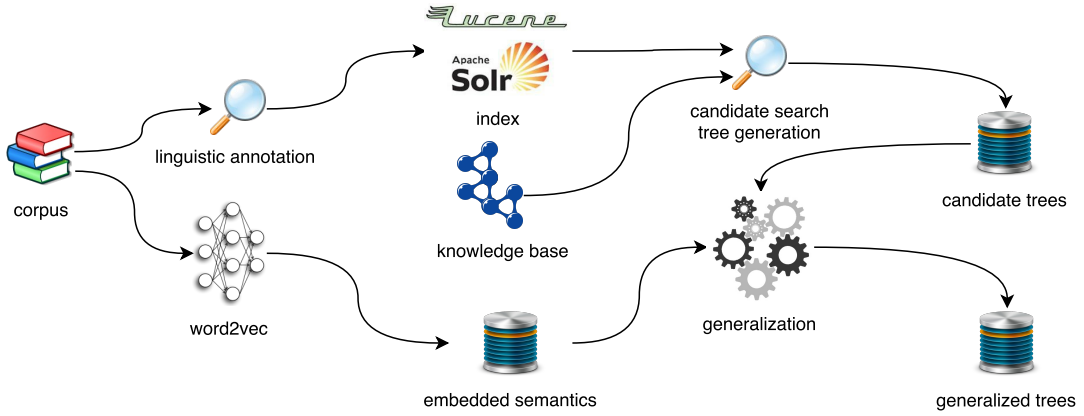

Abbildung 3: Architektur von Ocelot für RE

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 [49] 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.

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

[15] `https://github.com/dice-group/FOX/tree/master/documentation`

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

Literaturverzeichnis:

1. Manaal Faruqui and Sebastian Padó. Training and evaluating a german named entity recognizer with semantic generalization. In *Proceedings of KONVENS 2010*, Saarbrücken, Germany, 2010

2. 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

3. 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

4. L. Ratinov and D. Roth. Design challenges and misconceptions in named entity recognition. In *CoNLL*, 6 2009

5. Jason Baldridge. The opennlp project. *URL: http://opennlp.apache.org/index.html, (accessed 17 May 2017)*, 2005

6. 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

7. 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

8. 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

9. 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

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. 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

12. Daniel Gerber and Axel-Cyrille Ngonga Ngomo. Bootstrapping the linked data web. In *1st Workshop on Web Scale Knowledge Extraction @ ISWC 2011*, 2011

13. 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

14. 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

15. 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

16. 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

17. Axel-Cyrille Ngonga Ngomo, Norman Heino, Klaus Lyko, René Speck, and Martin Kaltenböck. Scms - semantifying content management systems. In *ISWC 2011*, 2011

18. 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

19. 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

Föderierte SPARQL Anfragen - QUETSAL Federation Suite CostFed, eine neuartige föderierte SPARQL Abfrage-Engine, war das Hauptergebnis von AP2. Die Engine ist ein Teil der QUETSAL Federation Suite. Die Datenquellenauswahl und die föderierte SPARQL Abfrage-Engine basieren auf einem Index, welcher aus den SPARQL-Endpunkten generiert wird. Die wichtigste Innovation hinter CostFed ist, dass es die ungleichmäßige Verteilung der Ressourcen zwischen verschiedenen SPARQL Endpunkten berücksichtigt, um eine effiziente Quellenauswahl und eine kostenbasierte Abfrageplanung durchzuführen. Unsere Experimente mit dem FedBench-Benchmark zeigen, dass Costfed 3 bis 121 mal schneller ist als der Stand der Technik.

Die Beantwortung komplexer Abfragen im *Web of Data* erfordert oft das Zusammenführen von Teilergebnissen aus verschiedenen Datenquellen. Die Optimierung von Abfragesystemens, die diese Art von Abfragen unterstützen, sogenannte *föderierte SPARQL Abfrage-Engine/ engl. federated query engines*, ist daher von zentraler Bedeutung für den effizienten und skalierbaren Einsatz von Semantic Web-Technologien. Aktuelle kostenbasierte föderierte SPARQL Abfrage-Engines [5, 18, 63] gehen davon aus, dass die RDF Ressourcen, die ein RDF Prädikat betreffen, gleichmäßig verteilt sind. Daher nutzen sie die *average selectivity*, um die Kardinalität von *Triple Pattern* zu schätzen. In Wirklichkeit sind die Ressourcen jedoch nicht gleichmäßig in den RDF-Datensätzen [10] verteilt. Unsere Analyse[16] des bekannten Benchmarks FedBench [45] bestätigt, dass die FedBench-Ressourcen nicht gleichmäßig verteilt sind. Der Nachteil der Verwendung von *average selectivity* für die Kardinalitätsschätzung mit *Triple Pattern* besteht darin, dass sie zu einer schlechten Kardinalitätsschätzung führen kann, wenn eine hochfrequente Ressource (d.h. eine Ressource, die in einer großen Anzahl von *Triple Pattern*n vorkommt) in diesem *Triple Pattern* verwendet wird. Somit kann die Abfrageplanung, wie von unserer föderierten SPARQL Abfrage-Engine vorgeschlagen, erheblich verbessert werden.

Evaluation: Wir haben FedBench [45] für die Evaluation verwendet, welche 25 Abfragen umfasst, von denen 14 (CD1-CD7, LS1-LS7) für föderierte SPARQL Abfrage-Engines geeignet sind (die anderen 11 Abfragen (LD1-LD11) sind für für Linked Data Federation Ansätze ausgelegt [43]).

Die Ausführungszeit der Abfrage wird häufig als Schlüsselmetrik für den Vergleich von föderierten SPARQL Abfrage-Engines verwendet. Dabei betrachten wir die Ausführungzeit der Abfrage als die Zeit, die erforderlich ist, um alle Ergebnisse aus dem Ergebnismengeniterator jeder Engine zu sammeln. Abbildung 4 zeigt die Laufzeitleistung der verschiedenen föderierten SPARQL Abfrage-Engines. Insgesamt übertrifft CostFed die anderen ausgewählten Systeme deutlich. Auf FedBench ist CostFed bei 11 von 14 Anfragen besser als FedX und übertrifft SPLENDID, ANAPSID und SemwGrow bei allen 14 Anfragen. Die durchschnittliche Laufzeit von CostFed über alle 14 FedBench-Abfragen beträgt nur 440ms, während FedX 7.468ms (d.h. die 16-fache Laufzeit von CostFed) benötigt, SPLENDIDs 5.3404ms (d.h, 121× langsamer als CostFed), ANAPSID 12.467ms (d.h. 28 mal die Zeit von CostFed) und SemaGrow's ist 1.203ms (d.h. 3 mal langsamer als CostFed). Da die Ausführungszeiten für die FedBench-Abfragen sehr

[16]FedBench Analyse: `https://github.com/AKSW/CostFed/tree/master/stats`

klein sind, d.h. weniger als 3 Sekunden mit CostFed betragen, ist die durchschnittliche Laufzeitleistung eines Systems stark beeinträchtigt, wenn eine bestimmte Abfrage zu lange dauert. So benötigt FedX beispielsweise 94.519 ms für die Ausführung von LS6, wodurch die Gesamtlaufzeitleistung im Vergleich zu CostFed stark reduziert wird. Wenn wir die LS6-Laufzeit entfernen, dann beträgt die durchschnittliche Laufzeit von FedX (über die restlichen 13 Abfragen) 771 ms (was immer noch 2 mal langsamer als CostFed ist).

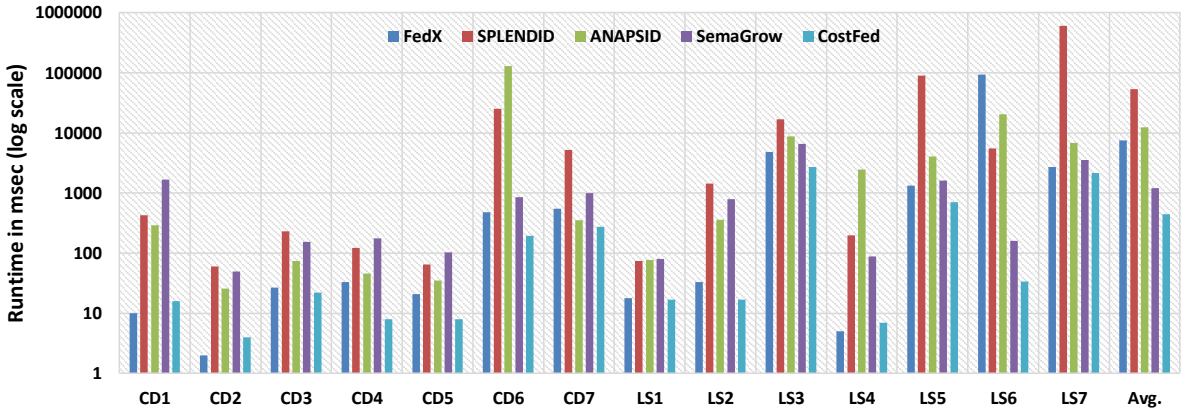

Abbildung 4: Vergleich der Query Ausführungszeiten auf FedBench.

Literaturverzeichnis:

1. Michael Schmidt, Olaf Görlitz, Peter Haase, Günter Ladwig, Andreas Schwarte, and Thanh Tran. Fedbench: a benchmark suite for federated semantic data query processing. In *ISWC*, 2011

2. Songyun Duan, Anastasios Kementsietsidis, Kavitha Srinivas, and Octavian Udrea. Apples and oranges: a comparison of rdf benchmarks and real rdf datasets. In *ACM SIGMOD*, 2011

3. Muhammad Saleem, Yasar Khan, Ali Hasnain, Ivan Ermilov, and Axel-Cyrille Ngonga Ngomo. A fine-grained evaluation of sparql endpoint federation systems. *SWJ*, 2015

4. Angelos Charalambidis, Antonis Troumpoukis, and Stasinos Konstantopoulos. Semagrow: Optimizing federated sparql queries. In *SEMANTICS*, 2015

5. Olaf Görlitz and Steffen Staab. Splendid: Sparql endpoint federation exploiting void descriptions. In *COLD at ISWC*, 2011

6. Xin Wang, Thanassis Tiropanis, and Hugh C Davis. Lhd: Optimising linked data query processing using parallelisation. In *LDOW at WWW*, 2013

Semantische Suche *SESSA* ist eine Keyword-basierte (dt. Stichwort-basierte) Entitäten-Suchmaschine. Die Implementierung basiert auf einer initialen Idee, welche auf dem NLIWOD Workshop 2014[17] von Prof. Ngonga vorgestellt wurde. Zuerst wird die Suchanfrage in ihre n-grams umgewandelt. Dies erlaubt uns zwischen Phrasen (wie dem Bigram 'Bill Gates') und einzelnen Wörtern (Unigrams wie 'wife' und 'birthplace') zu unterscheiden und eine Suche danach im Wörterbuch durchzuführen. Das Wörterbuch ordnet eine Menge von n-grams eine Menge von URIs zu. Seine Daten werden durch die Nutzung von Autoindex als REST-API und den Import von verschiedenen Dateitypen generiert. Diese Dateien beinhalten die Informationen des Wörterbuchs. Wir verwenden hauptsächlich das ttl-Format. Bei diesem Format beinhaltet jede Zeile ein RDF-Triple, das üblicherweise eine URI als Subjekt hat, sowie ein dazugehöriges Label (seine n-gram Darstellung) als Objekt. Die ttl-Dokumente erhalten wir bspw. von DBpedia oder Unternehmensdaten. Der Dokumentenverarbeitungsprozess ist in der Lage, diese Formate in die benötigte Datenstruktur zu transformieren. Diese Datenstruktur wird dann im Wörterbuch gespeichert, welches wir mit Apache Lucene[18] erstellen. Obwohl Lucene hauptsächlich als Dokumentenspeicher genutzt wird, kann es auch als einfaches Wörterbuch verwendet werden, was es uns ermöglicht die Levenshtein-Distanz zu nutzen. Diese kann z.B eine URI zu 'Schrödinger' finden, obwohl die Suchanfrage 'Schrödingers' (wie in Schrödingers Katze) lautete. In der Zukunft kann dafür Autoindex (siehe unten) als Wörterbuch verwendet werden. In der Nachbearbeitung zum Finden der URIs können diverse Filter auf die gefundenen URIs/Entitäten angewandt werden, die die Ergebnismenge verkleinern und so die Berechnungszeit im nächsten Schritt reduzieren können.

Das Wörterbuch wird nach URIs aus allen möglichen Kombinationen von n-grams aus der ursprünglichen Anfrage abgefragt. Die Antworten werden gespeichert und als initiale URIs zum Aufbau des Graphen weitergegeben. In diesem Prozess versucht der Algorithmus für jedes URI-Paar, dass keine Beziehung zueinander hat, das fehlende Triple-Element zu finden. Hierfür verwenden wir bspw. den SPARQL-Endpunkt von DBpedia oder den durch Costfed bereitgestellten, föderierten Endpunkt. Dieser Prozess wird solange wiederholt, bis alle Kombinationen der n-grams verbunden sind oder keine neuen URIs mehr gefunden werden können. Ist der Graph vollständig verbunden, ist das Ergebnis die Menge der URIs, die die ursprünglichen URIs verbindet. Sollten hingegen keine weiteren URIs mehr gefunden werden, sind diejenigen URIs das Ergebnis, die die meisten Wörter der Abfrage verbinden.

Wir haben SESSA mit dem QALD7 [55] Datensatz getestet. Die ersten Ergebnisse zeigen einen F-measure von ca. 0.15. Das niedrige Ergebnis kann großteils durch das noch sehr limitierte Wörterbuch erklärt werden, welches in DIESEL erarbeitet wurde. Diesen Teil des Algorithmus werden wir in der Zukunft verbessern, was zu besseren Ergebnissen führen wird. Der Code für SESSA kann im folgenden Repository gefunden werden: `https://github.com/dice-group/SESSA`.

[17] `https://svn.aksw.org/papers/2014/SESSA/public.pdf`
[18] `https://lucene.apache.org/core/`

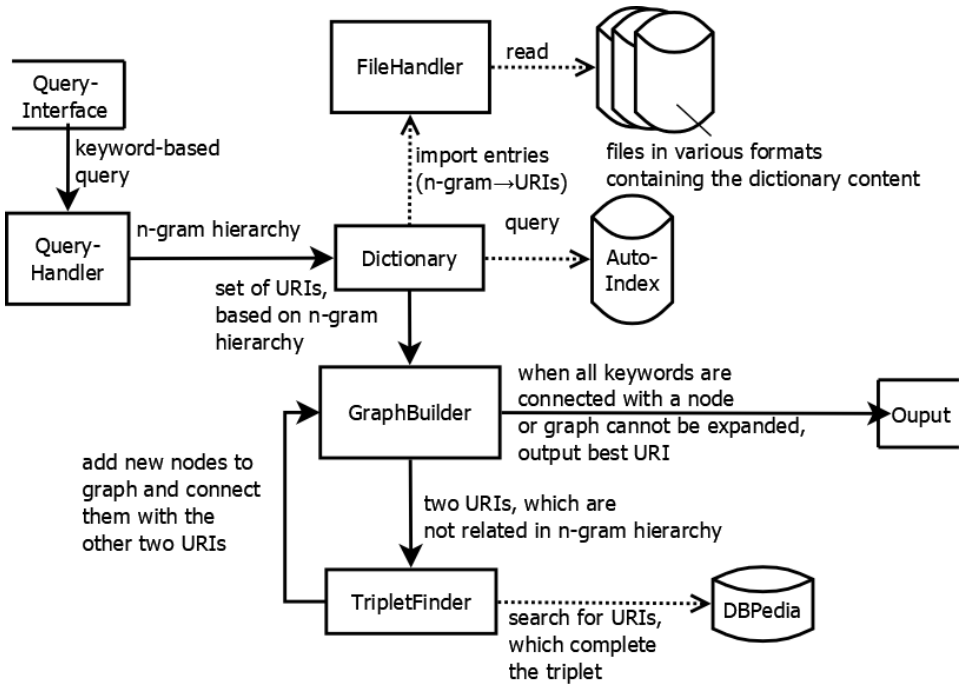

Abbildung 5: Architektur von SESSA

Literaturverzeichnis:

1. Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Bastian Haarmann, Anastasia Krithara, Michael Röder, and Giulio Napolitano. 7th open challenge on question answering over linked data (QALD-7). In *Semantic Web Evaluation Challenge*, pages 59–69. Springer International Publishing, 2017

Autoindex ist ein auf einer REST-API aufbauendes, effizientes Indexierungssystem, das darauf abzielt RDF-Speicherstrukturen oder SPARQL-Endpunkte für eine Datenabfrage zu indizieren. Es erlaubt Benutzern jegliche Art von RDF-Speicher zu indizieren und abzufragen, im Gegensatz zu anderen Diensten, die nur mit einer speziellen Art von RDF-Speicher arbeiten können. Autoindex basiert auf der Spring Data API[19], um eine praktische Schnittstelle für die Abfrage von Keywords und URLs bereitzustellen.

Normalerweise ermöglichen SPARQL-Endpunkte es Benutzern, den Triple Store im SPARQL-Abfrageformat abzufragen und die Ergebnisse über HTTP zurückzugeben. Diese Abfragen können bei der Stichwort- oder URI-basierten Suche nicht effizient durchgeführt werden. Autoindex schließt die Lücke, indem es die Spring Data API unterstützt und damit Elastic Search[20] in die Lage versetzt, auf komfortable Weise entweder über das Label oder die URI einer Entität zu suchen.

Die Spring Data API vereinfacht die Implementierung von Spring-basierten Anwendungen, die Datenzugrifftechnologien nutzen. Die Spring Data API bietet Schnittstellen zu, Elastic Search Repository, Elastic Search Operation und Mapping, was es erlaubt,

[19] `https://spring.io/projects/spring-data`
[20] `https://www.elastic.co/products/elasticsearch`

diese API Schnittstellen so zu erweitern, das sie standardmäßig CRUD-Operationen unterstützen. Darüber hinaus werden durch einfaches deklarieren von Methoden mit Namen in einem vorgegebenen Format Methodenimpelmentierungen generiert. Daher ist es nicht erforderlich, eine Implementierung der Repository-Schnittstelle zu schreiben.

Das Indizieren eines Endpunkts oder Speichers wird mittels Spring Data *ElasticsearchOperation* und *ElasticsearchRepository*-Schnittstelle realisiert.

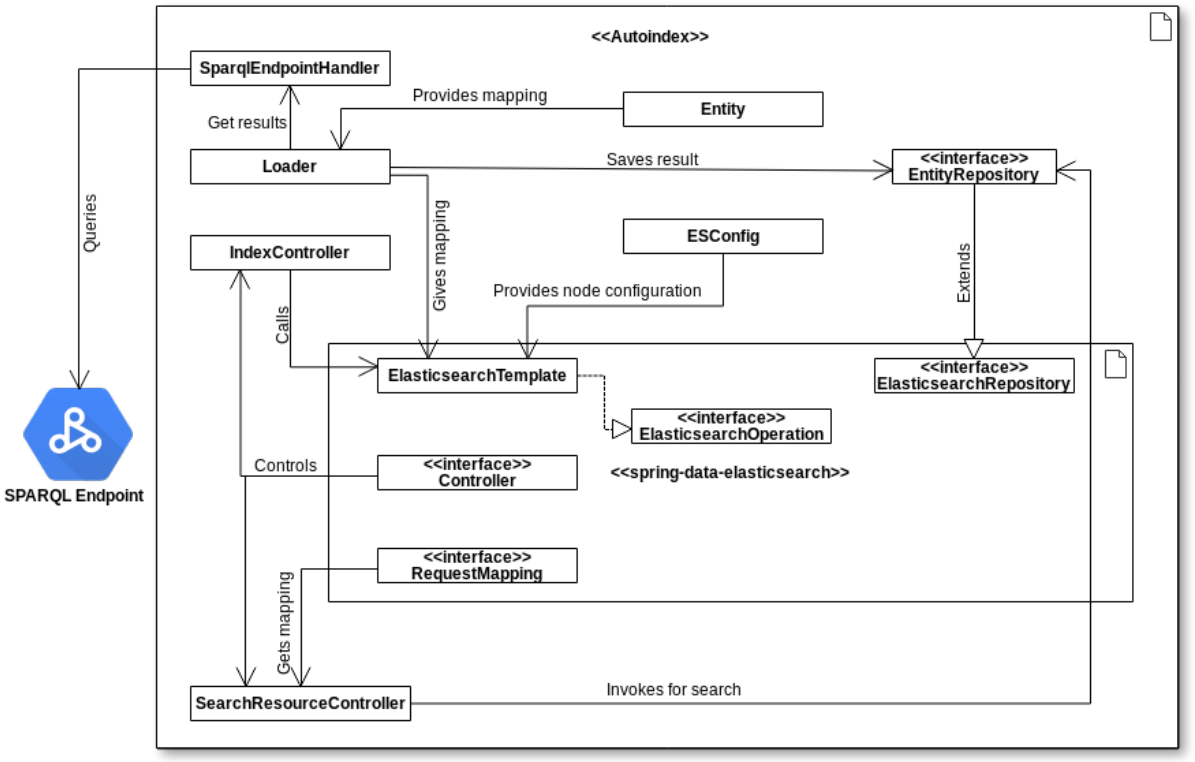

Abbildung 6: Architektur von Autoindex

Die *RequestMapping*-Schnittstelle ruft das Mapping für die zu indexierenden Daten in Elastic Search ab. Autoindex realisiert diese Operationen durch Objekte, um die Reihenfolge der Operationen sequentiell zu behandeln. Zuerst instanziiert die Loader-Klasse die *SparqlEndpointHandler*-Klasse, um das Ergebnis aller Dateninstanzen zu erhalten. Es verwendet das *Entität*-Objekt, um der Spring *ElasticsearchOperation*-Schnittstelle ein Mapping zur Verfügung zu stellen. Diese Schnittstelle verwendet das *ElasticsearchTemplate* als Client, um auf dem Elastic Search Server zu arbeiten. Autoindex stellt die Elastic Search-Konfigurationen zur Verfügung, die alle Informationen umfassen, die zum Aufbau der Knoten erforderlich sind. Die *EntityRepository*-Schnittstelle von Autoindex erweitert die *ElasticsearchRepository*-Schnittstelle und wird von der Klasse Loader realisiert, um alle Instanzen von Daten die vom *SparqlEndpointHandler* geholt werden, in den Elastic Search Server zu speichern.

Autoindex bietet auch eine einfache Weboberfläche für die Indizierung und Abfrage eines beliebigen RDF SPARQL-Endpunktes. Wenn es notwendig ist, einen Index

zu erstellen, benötigt die GUI eine Eingabe in Form einer funktionierenden SPARQL-Endpunkt-URI. Autoindex realisiert dies durch das *IndexController*-Objekt, das *ElasticsearchTemplate* mit Hilfe von Elastic Search operations aufruft. Für die Abfrage eines indizierten Endpunkts stellt die GUI zwei Eingabevarianten zur Verfügung. Entweder kann der Benutzer nach einem Suchbegriff (Label) suchen oder er kann eine URI angeben. Diese Operationen werden durch das Objekt *SearchResourceController* realisiert, das die Spring Data *Controller*-Schnittstelle verwendet. Das Ergebnis der Abfrage ist eine Liste der indizierten Entitäten, welche im Entity-Repository gefunden wurden. Eine Übersicht ist in Abbildung 6 dargestellt. Der Code für Autoindex kann im folgenden Repository gefunden werden: `https://github.com/dice-group/autoindex`.

Anwendungsstudie GENESIS Ziel der Anwendungsstudien war die Demonstration von Suchfunktionalitäten auf strukturierten und unstrukturierten Daten. Zu diesem Zweck wurde das User Interface GENESIS [13][21] erschaffen, welches einen nutzerfreundlichen Zugriff auf RDF Daten gewährleistet. GENESIS kann mit minimalem Aufwand auf jede Wissensbasis und jede Suchmaschine aufgesetzt werden. Die Architektur von GENESIS basiert auf Microservices und ermöglicht somit die Ersetzung einzelner Komponenten und die Erweiterung um weitere Funktionen. Die modulare Architektur der in DIESEL entwickelten Technologien erleichtert zudem diesen Prozess. Im Rahmen dieses Use Cases wurden folgende Funktionalitäten in GENESIS integriert, welche im Folgenden kurz vorgestellt werden.

1. Ähnlichkeitssuche über Entitäten
2. Semantische Stichwortsuche über DBpedia
3. Föderierte Suche über verschiedene Datenquellen
4. Extraktion von RDF Daten aus einer Datei mit unstrukturierten Archivdaten

1) Ähnlichkeitssuche über Entitäten: Für die Ähnlichkeitssuche wurden drei verschiedene Ansätze getestet. Der erste Ansatz nutzt eine SPARQL-Query, welche zunächst den spezifischsten Typ einer gegebenen Entität extrahiert und anschließend nach Entiäten mit dem selben Typ sucht. Der zweite, ebenfalls SPARQL-basierte Ansatz sucht dagegen die besten N Entitäten, welche Eigenschaften mit der Eingabeentität teilen. Der dritte Ansatz basiert auf einem vorab trainierten Vector Space Modell, welches auf Knowledge Graph Embeddings (KGE) basiert, bei denen Knoten und Kanten in einem Graphen Vektoren zugeordnet wird. Um die Skalierung für große Datensätze zu gewährleisten wurde als KGE das zeiteffiziente KG2Vec Embedding Model [46] verwendet, welches die skip-gram neural-network Architektur von Word2Vec anwendet indem jedes Triple als ein Satz mit Länge drei behandelt wird. Zur Reduzierung des Arbeitsspeicherbedarfs des gelernten Modells wurde der Incremental Principal Component Analysis (IPCA) Algorithmus zur Dimensionsreduktion verwendet. Somit konnte

[21] `https://github.com/dice-group/genesis`

der Arbeitsspeicherbedarg von 13GB auf 7GB reduziert werden, ohne die Performance signifikant zu beeinflussen. Die abschließende Evaluation auf zufällig ausgewählten Entitäten ergab, dass das KG2Vec basierte Verfahren bei vergleichbarer Qualität der Ergebnisse dreimal zeiteffizienter ist als die anderen beiden Verfahren.

2) Semantische Stichwortsuche über DBpedia: Für die semantische Suche wurde SESSA (siehe Abschnitt 2.1.2) in GENESIS integriert.
3) Föderierte Suche über verschiedene Datenquellen: Zur föderierten Suche wurde das Framework CostFed (siehe Kapitel 2.1.2) verwendet. Dabei wurden die folgenden Datensätze in CostFed geladen:

- Swiss Linked Archive Data: RDF Datensatz mit Schweizer Archivdaten, der über folgenden Endpunkt erreichbar ist: `http://data.alod.ch/query`.
- DBpedia: RDF Datensatz, welcher aus der Wikipedia Seiten extrahiert wurde. SPARQL Endpunkt: `http://dbpedia.org/sparql`.
- Jamendo: Ein großes französisches Repository mit Informationen zu Musik, welche unter der Creative Commons Lizenz verfügbar sind. SPARQL Endpunkt: `http://dbtune.org/jamendo/sparql/` .
- Europeana: Kollektion aus 58,112,930 Kunstwerken, Artefakten Büchern Filmen, Musik und vielem Weiterm aus europäischen Museen Galerien, Bibliotheken und Archiven. SPARQL Endpunkt: `https://pro.europeana.eu/resources/apis/sparql`.
- Linked Movies Dataset: Wissensbasis mit Informationen über Filme und Schauspieler. SPARQL Endpunkt: `http://www.linkedmdb.org/sparql`.

Abbildung 7 zeigt die von CostFed gefundenen Ergebnissse im GENESIS Interface für den Suchbegriff *Swiss*.
4) Extraktion von RDF Daten aus einer Datei mit unstrukturierten Archivdaten: Die Extraktion von Wissen aus unstrukturierten Daten beinhaltet zum einen die Erkennung von Entitäten (NER) und zum anderen das Linking der gefundenen Entitäten zu einer Wissensbasis. Zur Erkennung der Entitäten wurde ein Conditional Random Field (CRF) gelernt. Für das Linking wurde AGDISTIS [27] um Funktionalitäten erweitert, die das Linking zu mehreren Wissensbasen ermöglichen. Der Ansatz erreicht ein F1-measure von 93,2% für die Erkennung der Entitäten und 58,2% für das Linking der Entitäten zu den Wissensbasen DBpedia und der gemeinsamen Normdatei (GND)[22] der deutschen Nationalbibliothek. Der entstandene RDF Datensatz wurde in einen SPARQL Datensatz geladen, der ebenfalls über GENESIS abgefragt werden kann.

Evaluation von GENESIS: Zur Evaluation von GENESIS wurde zum zunächst eine kontrollierte Evaluation durch Vergleich mit den Systemen Wikipedia und dem Linked Data Browser Aemoo durchgeführt. Hierzu wurden drei unterschiedlich komplexe Fragen

[22] `https://www.dnb.de/DE/Standardisierung/GND/gnd_node.html`

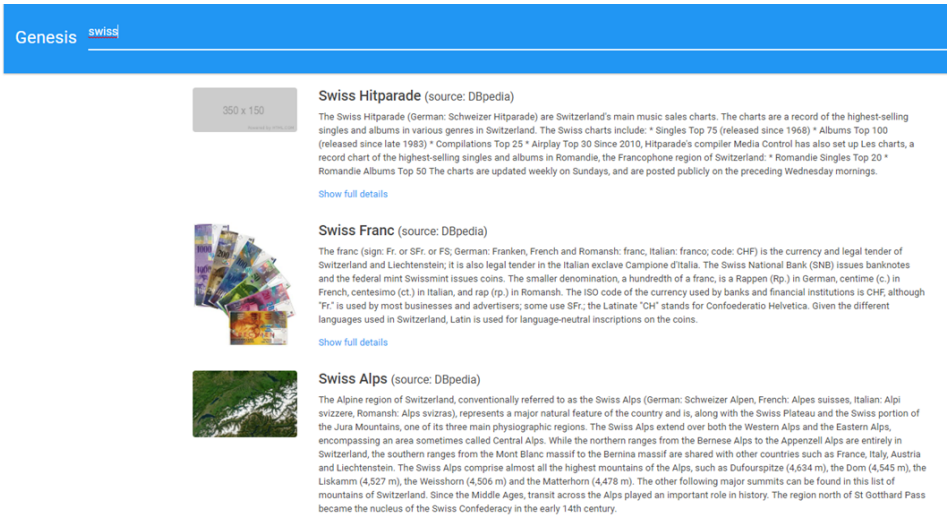

Abbildung 7: CostFed Ergebnisse im GENESIS Framework

ausgewählt, die Nutzer mithilfe der drei Systeme beantworten mussten. Insgesammt wurden 10 Nutzer aus unterschiedlichen Bereichen der Universität Leipzig ausgewählt. Bei dem Experiment wurden folgende Werte gemessen: (1) Zeit zur Beantwortung der Frage, (2) Länge des Klick Pfads und (3) Zufriedenheit der Nutzer mit dem Tool. Die Ergebnisse zeigen, das die Testpersonen mit GENESIS ebenso komfortabel arbeiten können, wie mit ihnen vertrauten Webtechnologien. Neben der kontrollierten Evaluation wurde ein System Usability Study[23] (SUS) mit 20 Nutzern durchgeführt das System erreicht einen SUS Score von 86,2, was bedeutet, das die getesteten Nutzer anderen Personen das System weiterempfehlen würden.

Literaturverzeichnis:

1. 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. ACM, 2017

2. Tommaso Soru, Stefano Ruberto, Diego Moussallem, Edgard Marx, Diego Esteves, and Axel-Cyrille Ngonga Ngomo. Expeditious generation of knowledge graph embeddings. 2018

3. 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

Vergleich mit angestrebten Zielen Die Universität Leipzig war an allen Arbeitspaketen beteiligt. Ziel von AP1 war eine Machbarkeitsstudie, eine Architektur sowie die

[23] https://en.wikipedia.org/wiki/System_usability_scale

Erarbeitung von Evaluationskriterien für DIESEL. Die Ergebnisse dieses Arbeitspaketes sind in Abschnitt 2.1.2 detailliert. In AP2 wurde eine Semantische Suche konzipiert und implementiert mittels der Systeme SESSA und AutoIndex. AP3 fokussierte sich auf die Erarbeitung einer föderierten SPARQL Suche. Hierfür wurden die Frameworks Quetsal und Costfed erweitert bzw. von Grund auf implementiert. In Arbeitspaket 4, wurden Systeme geschaffen und erweitert für die Extraktion von Entitäten und Relationen aus Volltexten. Schließlich wurde in AP5, eine Anwendungsstudie erstellt, um das Zusammenspiel der Komponenten zu evaluieren. Die Universität Leipzig unterstützte den Konsortialführer sowie alle Partner beim Projektmanagement durchgehend über den Projektzeitraum. Es wurden alle im Antrag genannten Ziele erreicht.

2.2 Zahlenmäßiger Nachweis

Insgesamt entsprechen die im Berichtszeitraum tatsächlich entstandenen Gesamtausgaben den ursprünglich geplanten Gesamtkosten. Geringfügige Abweichungen ergaben sich bei den Personalausgaben. Das Projekt wurde um zwei Monate kosteneutral verlängert, um den neuen Erfordernissen in der Projektarbeit aufgrund eines Partnerwechsels und dem darin begründeten Eurostars Amendment gerecht zu werden. Diese Verlängerung führte zu einem Mehrbedarf von 3,18% in der Position 0812, der durch einen Minderbedarf in der Position 843 gedeckt wurde. Alle Ausgaben im Projekt waren notwendig und wurden ausschließlich projektbezogen verwendet. Drei Positionen wurden insgesamt wie folgt genutzt.

1. **0812**: 226.639,50 Euro wurden für Personalkosten von wissenschaftlichen Mitarbeitern aufgebraucht. Hier ergab sich eine nur unwesentliche Abweichung von den geplanten Kosten aufgrund der notwendigen kostenneutralen Laufzeitverlängerung.

2. **0846**: 3.554,47 Euro wurden für Reise verwendet. Der wesentliche Unterschied zu den geplanten 10.200 Euro ergab sich daraus, dass mehrere Konferenzreisen im Jahr geplant waren, von denen jedoch nur eine pro Jahr genehmigt wurde. Ferner fanden aufgrund des Eurostars Amendments weniger Reisen als geplant statt.

3. **0850**: 15.096,74 Euro wurden für Investitionen genutzt. Hauptsächlich wurden mit Hilfe dieser Mittel zwei DIESEL Entwicklung-Server i.H.v. jeweils 7.499,95 Euro angeschafft. Diese werden nun für den Betrieb der Live Demonstratoren sowie für die nicht-wirtschaftliche Weiterentwicklung der DIESEL Technologien eingesetzt.

2.3 Notwendigkeit und Angemessenheit der geleisteten Arbeit

Die von der Universität Leipzig geleisteten Arbeiten erfolgten entsprechend der im Laufe von DIESEL spezifizierten Anforderungen [58]. Im Folgenden werden diese Anforderungen zusammengefasst. Es wird zusätzlich aufgezeigt, wie die DIESEL Technologien diese erfüllen.

Semantische Suche Für DIESEL war es notwendig, ein sematisches Suchsystem zu entwickeln, welches

1. natürlichsprachliche und Schlüsselwort-basierte semantische Anfragen beantworten,
2. über beliebingen Wissensbasen zur Anwendung kommen kann,
3. und welches eine hohe Genauigkeit auf Testdaten aufweist.

Durch das unterschiedliche technische Hintergrundwissen der Nutzer ergeben sich diverse Anforderungen an die Suchwerkzeuge für die semantische Suche. In DIESEL wurden daher verschiedene semantische Suchkomponenten genutzt. Um diesen Anforderungen (siehe WP 1) gerecht zu werden, sind vorhandene Suchalgorithmen angepasst und implementiert worden. Die Universität Leipzig hat mit SESSA (keywordbasierte Entitäten-Suchmaschine) und Autoindex (REST-API für jegliche Art von RDF-Speicher) eigene Suchsysteme entwickelt, so dass die gestellten Anforderungen an natürlichsprachige und Schlüsselwort-basierte Anfragen auf beliebige Wissensbasen und eine hohe Genauigkeit erfüllt werden.

Föderierte SPARQL Anfragen Für DIESEL war es notwendig, ein "Query Federation"Framework zu erschaffen, welches

1. mehrerer SPARQL Endpunkt gleichzeitig anfragen kann,
2. eine hohe Genauigkeit und
3. skalierbar ist, das heißt kleine Laufzeiten hat.

Das Quetsal Framework und der für DIESEL entwickelte Algorithmus Costfed bieten hier ein System für föderierte Anfragen, dass mit einer durchschnittlichen Genauigkeit (Precision) sowie Vollständigkeit (Recall) von mehr als 85% eines der effektivsten aber auch eines der effizientes Systeme der Welt ist.

Extraktion von Entitäten und Relationen aus Volltexten Für DIESEL war es notwendig ein Framework zu erschaffen, welches

1. Entitäten zu belibigen Wissensbasen verlinken kann,
2. eine RDF-fähige NIF[24] REST Schnittstelle implementierte und
3. eine durchschnittliche Genauigkeit (Precision) sowie Vollständigkeit (Recall) von mehr als 85% erreichte.

[24] `http://persistence.uni-leipzig.org/nlp2rdf/`

Die erste und zweite Anforderung ergaben sich aus der Notwendigkeit der Darstellung der Ergebnisse als RDF. Zu Beginn des Projektes stellte keins der existierenden nicht-kommerziellen Lösung für NER, NED und RE eine RDF-fähige Webschnittstelle zur Verfügung. Zusätzlich war die Genauigkeit sowie die Vollständigkeit der existierenden NER und NED Werkzeuge nicht zufriedenstellend, da die meisten industriellen Anwendungen Werte höher als 85% verlangen. Diese Werte wurden von FOX durch die intelligente Zusammenführung der Ergebnisse existierender Werkzeuge erreicht (siehe Abschnitt 2.1.2). Somit wurde nicht versucht, das "Rad neu zu erfinden", sondern es wurde auf existierende Technologien aufgebauen. Die Ergebnisse der Evaluation von FOX bestätigt, dass die intelligente Zusammenführung existierender Werkzeuge die geforderten Ziele bezüglich Genauigkeit und Vollständigkeit erreicht.

Anwendungsstudien Für DIESEL war es notwendig einen industriellen Use Case zu schaffenm welcher den Einsatz von sowohl

1. semantischer Suche als auch
2. föderierter Suche
3. auf strukturierten
4. und unstrukturierten Daten erfolgreich demonstriert.

Die erste Anforderung ergibt sich aus der generellen Notwendigkeit für Unternehmen große Menge an Daten aus unterschiedlichen Quellen hinsichtlich unterschiedlichster Kriterien durchsuchen zu können. Darauf aufbauend ergibt sich die zweite Anforderung daraus, dass Daten häufig aus unterschiedliche Quellen auf effiziente Weise kombiniert werden müssen, um neue Informationen zu schaffen. Die zweite und dritte Anforderung ergibt sich aus der Heterogenität der Daten. Als Anwendungsstudie wurde das GENESIS-Framework entwickelt, dem Nutzer einen zentralen Zugriff auf Suchergebnisse aus unterschiedlichsten Suchapplikatioen bietet. Die erste Anforderung wurde durch die Integration der Ähnlichkeissuche und der semantischen Suche in das GENESIS-Framework erreicht. Die zweite Anforderung wurde durch die Integration von CostFed in GENESIS erfüllt, was eine föderierte Suche über insgesamt 5 verschiedene Wissensbasen ermöglicht. Die letzten beiden Anforderungen wurden dadurch erfüllt, dass sowohl strukturierte Wissensbasen wie bspw. DBpedia als auch unstrukturierte Daten wie eine Datei mit Archivdaten in den Use Case integriert wurden. DIESEL ist somit für den industriellen Einsatz vollständig vorentwickelt.

2.4 Voraussichtlicher Nutzen

Alle vom ULEI im Rahmen von DIESEL entwickelten Technologien stehen der Öffentlichkeit als Open-Source oder Open-Service Frameworks zur Verfügung. Wir planen u.a. eine um DIESEL-erweiterte Version von Wikipedia zu veröffentlichen, um das Nutzen und die Nutzbarkeit unserer Technologien noch klarer zu demonstrieren. Zusätzlich zum

gesamten DIESEL Framework können die einzelnen Komponenten von DIESEL unabhängig von einander genutzt werden.

SESSA und Autoindex werden von der Arbeitsgruppe von Prof. Ngonga in Zukunft als Ausgangspunkt für weitere Suchmechanismen genutzt. Diese Mechanismen werden sowohl in der Lehre im Rahmen der Vertiefung 'Data Science' ausgebaut als auch in der weiteren Forschung zur Anwerbung und Durchführung neuer Projekte genutzt. In der Zukunft ist ebenfalls angedacht die erarbeiteten Mechanismen zusammen mit Industriekunden zu nutzen, um strukturierte Datenmenge, welche durch KI Prozesse entstanden sind, einfacher durchsuchbar zu machen.

Quetsal und Costfed dienen der dörderierung von SPARQL-Anfragen über viele SPARQL-Endpunkte hinweg und ermöglichen so eine effektive und effiziente Query-Ausführung für verteilte RDF Triple Stores. Dies ist insbesondere bei Unternehmen wichtig bei denen sich im Laufe der Zeit diverse Datenquellen angesammelt haben. Diese Datenseen (engl. Data Lakes) können über die Transformation der Daten nach RDF nun via der hier entwickelten Föderierungsmechanismen angefragt und so gemeinsam genutzt werden anstatt als Datensilos zu existieren. Es ist geplant die noch offenen Forschungsfragestellungen von Costfed und Quetsal als Grundlage weiterer Eurostarsanträge bzw. DFG Anträge zu beantworten. Daneben ist die Kommerzialisierung der Systeme via Lizensierungsmodellen geplant.

Der Hauptnutzen von FOX besteht in der Extraktion von strukturierten Daten aus Text. Mit Hilfe dieser Funktionalität können Daten aus dem Document Web so angereichert werden, dass sie von Menschen und Maschinen besser verarbeitet werden können. Somit liefert FOX einen der wesentlichen Bausteine zur Umsetzung des Semantic Webs und kann von Bloggern, Webseiten-Betreibern und Suchmaschinen genutzt werden. Das FOX Projekt wurde bereits erfolgreich in mehrere Lösungen eingebettet. Zum Beispiel nutzt GeoLift[25] 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. Ferner wird FOX bereits von mehreren Einrichtungen verwendet (z.B. von HL Komm, der University of Oxford und der George Washington University).

2.5 Fortschritt bei anderen Stellen

Im Rahmen der DIESEL 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 Stichwortsuche

Semantische Stichwortsuchalgorithmen (engl. semantic keyword search algorithms) erfreuen sich immer noch einer regen Beliebtheit in der Forschungscommunity. So wurde

[25] `http://github.com/AKSW/GeoLift`

das EU COST Projekt Keystone (2014-2017) ins Leben gerufen.[26] Neben den in dem White Paper dargestellten Arbeiten[27] sind uns seit 2017 keine neuen Arbeiten außerhalb unserer Forschungsgruppe bekannt, die Ähnlichkeit zu den in DIESEL avisierten Technologien aufzeigen. Semantische Stichwortsuche ist in der Industrie ebenfalls angekommen und in mehreren uns bekannten Produkten integriert, so bspw. in metaphacts metafactory. Allerdings sind Systeme anderer Anbieter meist nur im Verbund mit anderen Produkten erhältlich und anders als DIESEL Komponenten nicht alleine lauffähig.

2.5.2 Föderierte Anfragen

Zuletzt wurden verschiedene, föderierte SPARQL-Abfragesysteme wie Odyessey, Lusail und SemaGrow entwickelt. Odyesey verwendet verteilte Charecteristic Sets und Charecteristic Pair (CP)-Statistiken, um Kardinalitäten zu schätzen. Anschließend erstellt es mit Hilfe dynamischer Programmierung Query-Ausführungspläne. Semagrow implementiert ein kostenbasiertes, föderiertes Abfragesysteme. Die Abfrageplanung basiert auf VoID-Statistiken über Datensätze. SemaGrow implementiert Bind, Hash und Merge Joins und die Auswahl eines dieser Joins basiert auf der Kostenkalkulation, um den gewünschten Join-Vorgang durchzuführen. Es verwendet eine reaktive Modell zum Abrufen der Ergebnisse der Verbindungen sowie einzelner Triple-Pattern. Schließlich ist Lusail eine indexfreie Engine, die Datenlokalität nutzt, um Query-Ausführungspläne zu erstellen. Die Kardinalitäten von Joins werden zunächst mit Hilfe von SPARQL Count Abfragen berechnet. Diese Informationen werden später bei der Join-Bestellung verwendet.

2.5.3 Wissensextraktion

Während der Projektlaufzeit wurde eine Vielzahl von Wissensextraktions-Werkzeugen entwickelt. Frameworks wie DBpedia Spotlight[28] und Wikipedia Miner[29] 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[30] und der Illinois Named Entity Tagger[31] werden ständig weiterentwickelt und erzielen immer bessere Precision und Recall-Werte [59]. Da sie in FOX eingebettet sind, bedeutet dies, dass FOX fortwährend besser wird. Kommerzielle Lösungen wie Alchemy[32] and Extractiv[33] können genauso in FOX eingebettet werden. Details können [35] entnommen werden.

[26] http://www.keystone-cost.eu/
[27] http://www.keystone-cost.eu/keystone/wp-content/uploads/2015/01/WhitePaper_first.pdf
[28] http://spotlight.dbpedia.org
[29] http://wikipedia-miner.cms.waikato.ac.nz/
[30] http://nlp.stanford.edu/software/CRF-NER.shtml
[31] http://cogcomp.cs.illinois.edu/page/software_view/4
[32] http://www.alchemyapi.com/api/
[33] http://extractiv.com/

2.6 Erfolgte und geplante Veröffentlichungen

2.6.1 Erfolgte Veröffentlichungen

Im Rahmen von DIESEL 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://diesel-project.eu`.

1. Matthias Wauer, Mohamed Sherif, Muhammad Saleem, Olaf Hartig, Ricardo Usbeck, Ruben Verborgh, and Axel-Cyrille Ngonga Ngomo, editors. *Proceedings of the 3rd International Workshop on Geospatial Linked Data and the 2nd Workshop on Querying the Web of Data co-located with 15th Extended Semantic Web Conference (ESWC 2018), Heraklion, Greece, June 3, 2018*, volume 2110 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2018

2. Ricardo Usbeck, Michael Röder, Muhammad Saleem, and Axel-Cyrille Ngonga Ngomo. Diesel – distributed search over large enterprise data. In *ESWC, EU networking session*, 2016

3. 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

4. Ricardo Usbeck, Michael Röder, Peter Haase, Artem Kozlov, Muhammad Saleem, and Axel-Cyrille Ngonga Ngomo. Requirements to modern semantic search engines. In *KESW*, 2016

5. Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Jin-Dong Kim, Key-Sun Choi, Philipp Cimiano, Irini 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

6. 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

7. Ricardo Usbeck, Jonathan Huthmann andNico Duldhardt, and Axel-Cyrille Ngonga Ngomo. Self-wiring question answering systems. *CoRR*, 1611.01802, 2016

8. Mohamed Ahmed Sherif andAxel Cyrille Ngonga Ngomo Abdullah Fathi Ahmed. Radon2: A buffered-intersection matrix computing approach to accelerate link discovery over geo-spatial rdf knowledge bases (oaei2018 results). In *Proceedings of Ontology Matching Workshop 2018*, 2018

9. Muhammad Saleem, Ricardo Usbeck, Michael Röder, and Axel-Cyrille Ngonga Ngomo. Sparql querying benchmarks. In *Tutorial at ISWC*, 2016

10. Muhammad Saleem, Ali Hasnainb, and Axel-Cyrille Ngonga Ngomo. Largerdfbench: A billion triples benchmark for sparql endpoint federation. In *Journal of Web Semantics (JWS)*, 2017

11. 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

12. Muhammad Saleem. Efficient source selection for sparql endpoint query federation. In *PhD thesis at AKSW, University of Leipzig Germany*, 2016

13. Michael Röder, Ricardo Usbeck, and Axel-Cyrille Ngonga Ngomo. Gerbil - benchmarking named entity recognition and linking consistently. *Semantic Web*, 9(5):605–625, 2018

14. Axel-Cyrille Ngonga Ngomo and Muhammad Saleem. Federated query processing: Challenges and opportunities. In *Keynote at PROFILES at Extended Semantic Web Conference (ESWC)*, 2016

15. 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

16. 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

17. 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

18. Diego Moussallem, Marcos Zampieri Thiago Castro Ferreira, Maria Claudia Cavalcanti, Geraldo Xexéo, Mariana Neves, and Axel-Cyrille Ngonga Ngomo. Rdf2pt: Generating brazilian portuguese texts from rdf data. In *The 11th edition of the Language Resources and Evaluation Conference, 7-12 May 2018, Miyazaki (Japan)*, 2018

19. Diego Moussallem, Mohamed Ahmed Sherif, Diego Esteves, Marcos Zampieri, and Axel-Cyrille Ngonga Ngomo. Lidioms: A multilingual linked idioms data set. In *The 11th edition of the Language Resources and Evaluation Conference, 7-12 May 2018, Miyazaki (Japan)*, 2018

20. Jin-Dong Kim, Christina Unger, Axel-Cyrille Ngonga Ngomo, André Freitas, Young gyun 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

21. Jin-Dong Kim, Christina Unger, Axel-Cyrille Ngonga Ngomo, André Freitas, Young-Gyun 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 and Demonstrations and Industry Tracks co-located with 16th International Semantic Web Conference (ISWC 2017), Vienna, Austria, October 23rd - to - 25th, 2017.*, 2017

22. Mohamed Ahmed Sherif andAxel Cyrille Ngonga Ngomo Kevin Dreßler. Radon results for oaei 2017. In *Proceedings of Ontology Matching Workshop 2017*, 2017

23. Ali Hasnainb, Muhammad Saleem, Axel-Cyrille Ngonga Ngomo, and Dietrich Rebholz-Schuhmann. Extending largerdfbench for multi-source data at scale for sparql endpoint federation. In *ISWC Satellite, SSWS*, 2018

24. 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

25. Daniel Gerber, Diego Esteves, Jens Lehmann, Lorenz Bühmann, Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, and René Speck. Defacto - temporal and multilingual deep fact validation. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2015

26. 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

27. Ivan Ermilov and Axel-Cyrille Ngonga Ngomo. Taipan: Automatic property mapping for tabular data. In *Proceedings of 20th International Conference on Knowledge Engineering and Knowledge Management (EKAW'2016)*, 2016

28. Kevin Dreßler and Axel-Cyrille Ngonga Ngomo. On the efficient execution of bounded jaro-winkler distances. In *Semantic Web Journal*, 2016

29. Kevin Dreßler and Axel-Cyrille Ngonga Ngomo. Time-efficient execution of bounded jaro-winkler distances. In *Proceedings of Ontology Matching Workshop*, 2014

30. 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

31. Ethem Cem Ozkan, Muhammad Saleem, Erdogan Dogdu, and Axel-Cyrille Ngonga Ngomo. Upsp: Unique predicate-based source selection for sparql endpoint federation. In *PROFILES at Extended Semantic Web Conference (ESWC)*, 2016

32. *Joint proceedings of the second RDF stream processing and the querying the web of data workshops*, 2017

33. 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

34. 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 DIESEL-Ergebnisse werden aktiv weitergepflegt und stetig erweitert. In dieser Hinsicht geplant ist eine weitere Publikation zu Costfed worin weitere überwachte maschinelle Lernverfahren im Rahmen ihres Einsatzes im Aufbau von föderierten Abfragesystemen verglichen werden sowie weitere Artikel zu Erkenntnissen aus Benchmarks. Geplant sind ebenfalls weitere Veröffentlichungen im Bereich Wissensextraktion sowie Suche und Autovervollständigung.