Linked Data Based Knowledge Provisioning

by

Ing. Milan Dojchinovski

DISSEPTION THESIS

submitted in partial fulfillment of the requirements for the degree of

DOCTOR (Ph.D.)

at the Faculty of Information Technology,
Czech Technical University in Prague

Doctoral study programme: Informatics
Department of Software Engineering
Prague, August 2018
Supervisor:

doc. Ing. Tomáš Vitvar, Ph.D.
Department of Software Engineering
Faculty of Information Technology
Czech Technical University in Prague
Thákurova 9
160 00 Prague 6
Czech Republic

Copyright © 2018 Ing. Milan Dojchinovski
Abstract and contributions

The ultimate goal of the World Wide Web is to enable efficient publishing, sharing and consumption of information. Until recently, the Web has been primarily used as a medium for publishing and sharing documents, also known as the Web of Documents. In the last decade, we could witness an evolution from a Web of Documents to a Web of Data. This noticeable transition has been primarily supported by the Linked Data paradigm, which has emerged as a simple mechanism for publishing, sharing and consuming data, information and knowledge.

Linked Data has already attracted many different communities from different domains, such as the life sciences, government, geography and linguistics, which have published huge amount of data according to the Linked Data principles. In spite of this increase, several crippling problems are starting to surface. First, although there are many domains which are well covered in the Linked Open Data cloud, there are still domains which are marginally, or not covered at all. Second, while Linked Data aims at publishing integrated knowledge, yet huge amount of valuable knowledge is left hidden in an unstructured format. In order to realize the vision of the Semantic Web, there is need to extract and transform this hidden information into structured data, and integrate it into the Linked Data space. And third, due to the huge and diverse amount of information published as Linked Data, the retrieval of information from the LOD cloud demands significant amount of effort; standard retrieval mechanisms require background knowledge on the schema and have query expressivity limitations.

This dissertation thesis addresses the aforementioned problems related to Linked Data, and in particular: *domain-specific data acquisition, semantization and Linked Data publishing* - we deal with data acquisition and semantization for the Web services domain, *knowledge extraction from unstructured data and its integration with Linked Data* - we exploit named entity recognition and entity linking techniques to extract relevant information and integrate it with Linked Data, and *personalized retrieval of Linked Data* - we develop a graph-based method for personalized retrieval of Linked Data resources.

The main contributions of the dissertation thesis can be summarized as follows:
1. A Linked Data dataset with semantic Web API descriptions, supported with a lightweight ontology and a survey on the usefulness of the dataset.

2. An open-source system for named entity recognition and entity linking, supported with a method for learning entity salience, ground-truth datasets for named entity recognition and entity salience, and experimental evaluations.

3. A method for personalized retrieval of Linked Data which outperforms traditional personalized and non-personalized methods, an evidence that semantics improve the accuracy of the recommendations, an evidence on the impact of the resource informativeness on the accuracy, serendipity and diversity, and an evidence on the trade-off between accuracy, serendipity and diversity of the recommendations.

Keywords:
Semantic Web, Linked Data, Data Acquisition, Web APIs, Information Extraction, Named Entity Recognition, Entity Linking, Linked Data Recommenders
Acknowledgements

First of all, I would like to express my gratitude to my dissertation thesis supervisor doc. Ing. Tomáš Vitvar, Ph.D. for giving me the opportunity to pursue PhD. He has been a source of encouragement and insight during my research, helped me with numerous problems and professional advancements and provided valuable feedback on the contents of the thesis.

Special thanks go to the head of the Department of Software Engineering, Ing. Michal Valenta, Ph.D., who supported me since the very beginning of my PhD and provided a pleasant and flexible environment for my research. Thanks also to the board of the Faculty of Information Technology for giving me the opportunity to enroll the doctoral study programme besides my little knowledge of the Czech language. I truly appreciate it!

I would also like to thank Prof. Vojtěch Svátek and Ing. Tomáš Kliegr, Ph.D. for giving me the opportunity to participate in the LinkedTV and LOD2 EU projects. These two projects greatly helped me to improve my research skills and get in touch with the Semantic Web and DBpedia communities.

Very special thanks also go to Dr.-Ing. Sebastian Hellmann for giving me the great opportunity to be part of the AKSW research team and the KILT family. Moreover, I would like to thank Sebastian for his constant support, motivation, guidance, stimulating discussions, keen observations and his confidence in me that helped me to progress well in my work. I would also like to thank the colleagues at AKSW for their support and engaging discussions throughout the past several years.

Thanks also to all reviewers of my papers for their time and valuable feedback, which greatly helped me to improve the quality of my work. Thanks also to all my co-authors for believing in my ideas and their work so that this thesis could be completed.

My research would be more difficult without a funding, hence I would like to thank the European Commission for the partial funding of my research via the FP7 LinkedTV (GA 287911), FP7 LOD2 (GA 257943) and H2020 FREME (GA 644771) projects, the Grant Agency of the Czech Technical University in Prague (grant No. SGS12/093/0HK3/1T/18, SGS13/100/OHK3/1T/18 and SGS14/104/OHK3/1T/18) and the International Visegrad Fund for the partial funding of my research via the In-Coming Scholarship (No. 51200853
and 51100570). Also, I would like to thank the Department of Software Engineering and
the Faculty of Information Technology for providing funding for my research.

Big thanks to all my relatives and friends for their support, care and the very friendly
welcome during my visits back home. These visits helped me a lot to renew my energy
which I needed to keep myself in shape and perform well in my work.

Most importantly, I would like to thank my parents, Ljubica and Dragi, and my sister
Marija, for their endless love and support during the studies. Ви благодарим за целосната
поддршка и пожртвуваност за време на студиите. Без вас ќе беше многу тешко да ги
оставам моите планови. Благодарам и за многите пакети поли со специалитети,
а пред сè за многу вкусенот ајвар, јуфки и сирење. Без овие три состојки ќе беше
многу тешко да се остане овие неколку години. Very big thanks goes also to my closest
family members, Hana, Filip and Anna, for their support, love, motivation, understanding
and time given for the work on the PhD. Hanka, thanks a lot for your patience, for the
many encouragement words, and especially, for always being there to listen about my new
ideas, research problems and ongoing experiments. Филипче, фала за даденото време и
мотивација да ја завршам докторската. Анче, ти ми беше последната и најважната
мотивација која ми требаше да го завршам ова поглавје, фала ти!

Milan
Prague
November 16th, 2017
Dedication

To my beloved family
## Contents

<table>
<thead>
<tr>
<th>Abbreviations</th>
<th>xix</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Motivation Scenarios</td>
<td>2</td>
</tr>
<tr>
<td>1.2 Problem Statement</td>
<td>4</td>
</tr>
<tr>
<td>1.2.1 Knowledge Acquisition and Semantization</td>
<td>6</td>
</tr>
<tr>
<td>1.2.2 Knowledge Extraction and Integration</td>
<td>8</td>
</tr>
<tr>
<td>1.2.3 Personalized Knowledge Retrieval</td>
<td>10</td>
</tr>
<tr>
<td>1.3 Related Work and Previous Results</td>
<td>11</td>
</tr>
<tr>
<td>1.4 Contributions of the Thesis</td>
<td>14</td>
</tr>
<tr>
<td>1.5 Research Methodology</td>
<td>16</td>
</tr>
<tr>
<td>1.6 Structure of the Thesis</td>
<td>18</td>
</tr>
<tr>
<td>2 Background and State-of-the-Art</td>
<td>19</td>
</tr>
<tr>
<td>2.1 Theoretical Background</td>
<td>19</td>
</tr>
<tr>
<td>2.1.1 Semantic Web</td>
<td>19</td>
</tr>
<tr>
<td>2.1.2 Knowledge Extraction and Integration</td>
<td>27</td>
</tr>
<tr>
<td>2.1.3 Recommender Systems</td>
<td>30</td>
</tr>
<tr>
<td>2.2 Previous Results and Related Work</td>
<td>34</td>
</tr>
<tr>
<td>2.2.1 Semantic Web Services</td>
<td>34</td>
</tr>
<tr>
<td>2.2.2 Named Entity Recognition and Entity Linking</td>
<td>41</td>
</tr>
<tr>
<td>2.2.3 Linked Data Recommenders</td>
<td>48</td>
</tr>
<tr>
<td>3 Acquisition and Semantization of Web API Descriptions</td>
<td>53</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>55</td>
</tr>
<tr>
<td>3.2 The Data Source</td>
<td>56</td>
</tr>
<tr>
<td>3.3 The Ontology</td>
<td>56</td>
</tr>
<tr>
<td>3.4 The Linked Web APIs Dataset - Coverage, Availability and Maintenance</td>
<td>59</td>
</tr>
<tr>
<td>3.4.1 Coverage and Availability</td>
<td>59</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Contents

4 Knowledge Extraction and Integration with Salient Linked Entities

4.1 Named Entity Recognition and Linking

4.1.1 Introduction

4.1.2 Approach

4.1.3 Implementation

4.1.4 The Web API

4.1.5 Evaluation and Validation

4.1.6 Achieved Impact and Use Cases

4.1.7 Discussion and Future Work Directions

4.1.8 Summary

4.2 Learning Entity Salience

4.2.1 Introduction

4.2.2 Entity Salience Identification Methodology

4.2.3 Crowdsourcing Entity Salience Annotations

4.2.4 Experiments

4.2.5 Discussion

4.2.6 Summary

4.3 Evaluation Framework for Named Entity Recognition

4.3.1 Available Datasets

4.3.2 News and Tweets Evaluation Datasets

4.3.3 GATE Evaluation Framework

4.3.4 Availability and License

4.3.5 Evaluation Results

4.3.6 Summary

4.4 Chapter Summary

5 Personalized Retrieval of Linked Data

5.1 Introduction

5.2 Personalised Resource Recommendation
1.1 An illustration of the thesis focus on “Linked Data based Knowledge Provisioning” .............................................. 5  
2.1 A simple RDF graph example .......................................................... 21  
2.2 Excerpt from the LOD Cloud. A modified version of an image retrieved from http://lod-cloud.net/ .................................................. 25  
3.1 The Linked Web APIs Ontology .......................................................... 57  
3.2 Web API utilization over time .......................................................... 64  
3.3 Popularity of HTTP APIs and SOAP based services over time ............ 65  
3.4 Usefulness of the dataset as seen by the consumers (left) and providers (right) ........................................... 66  
3.5 Will consumers (left) and providers (right) consider the dataset in the near future ........................................... 67  
4.1 Architecture overview .......................................................... 77  
4.2 Schematic overview of the methodology for identification of salient entities 111  
5.1 Excerpt from the Linked Web APIs dataset with resource context graphs with context distance of 3 ........................................... 134  
5.2 Caption for LOF .......................................................... 143  
5.3 Trade-off between serendipity and accuracy studied @ top 5, 10, 15 and 20 145
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Query results from the SPARQL query from Listing 2.2.</td>
<td>24</td>
</tr>
<tr>
<td>2.2</td>
<td>Summary of Web Service Models and Data Sources. Data collected as of July 2017.</td>
<td>39</td>
</tr>
<tr>
<td>2.3</td>
<td>Named Entity Recognition and Entity Linking Systems.</td>
<td>47</td>
</tr>
<tr>
<td>2.4</td>
<td>Summary of Linked Data Recommenders and Exploratory Search Systems.</td>
<td>51</td>
</tr>
<tr>
<td>3.1</td>
<td>Details of the Linked Web APIs dataset.</td>
<td>59</td>
</tr>
<tr>
<td>3.2</td>
<td>Number of linked resources per type and dataset.</td>
<td>61</td>
</tr>
<tr>
<td>4.1</td>
<td>Evaluation results for spotting of named entities.</td>
<td>80</td>
</tr>
<tr>
<td>4.2</td>
<td>Evaluation results for classification of named entities.</td>
<td>80</td>
</tr>
<tr>
<td>4.3</td>
<td>Evaluation results for linking of named entities.</td>
<td>80</td>
</tr>
<tr>
<td>4.4</td>
<td>Evaluation results for spotting of common entities.</td>
<td>81</td>
</tr>
<tr>
<td>4.5</td>
<td>Evaluation results for classification of common entities.</td>
<td>81</td>
</tr>
<tr>
<td>4.6</td>
<td>Evaluation results for linking of common entities.</td>
<td>81</td>
</tr>
<tr>
<td>4.7</td>
<td>Size metrics for the S&amp;V and RBB datasets.</td>
<td>82</td>
</tr>
<tr>
<td>4.8</td>
<td>Evaluation results for Dutch. An experiment type A2KB - named entities only.</td>
<td>83</td>
</tr>
<tr>
<td>4.9</td>
<td>Evaluation results for Dutch. An experiment type A2KB - named and common entities.</td>
<td>84</td>
</tr>
<tr>
<td>4.10</td>
<td>Evaluation results for Dutch. An experiment type A2KB - common entities only.</td>
<td>84</td>
</tr>
<tr>
<td>4.11</td>
<td>Evaluation results for German. An experiment type A2KB - named entities only.</td>
<td>84</td>
</tr>
<tr>
<td>4.12</td>
<td>Evaluation results for German. An experiment type A2KB - named and common entities.</td>
<td>85</td>
</tr>
<tr>
<td>4.13</td>
<td>Evaluation results for German. An experiment type A2KB - common entities only.</td>
<td>85</td>
</tr>
<tr>
<td>4.14</td>
<td>Overall performance of all the runs.</td>
<td>90</td>
</tr>
<tr>
<td>4.15</td>
<td>Comparison of the highest and median $B^3F_1$ scores achieved at the challenge with the highest score achieved by our submissions.</td>
<td>91</td>
</tr>
</tbody>
</table>
4.16 Results achieved by each team with their best performing run. Brief description is provided only for the approaches which are accompanied with a TAC 2013 paper.

4.17 Results for queries targeting GPE (Geographical-Political) entities.

4.18 Results for the queries targeting ORG (Organization) entities.

4.19 Results for queries targeting PER (Person) entities.

4.20 Results for queries from the English discussion forum documents selected from the BOLT Phase 1 forum data.

4.21 Results for the queries from the Web documents from various GALE web collections.

4.22 Results for queries from the Newswire documents from the English Gigaword Fifth Edition.

4.23 Results for queries targeting entities which are not present in the KB.

4.24 Results for queries targeting entities which are present in the KB.

4.25 Results from the entity mention detection evaluation - Strong Mention Match metric.

4.26 Results from entity linking evaluation - Strong All Match metric.

4.27 Results from the entity classification evaluation - Strong Typed Mention Match metric.

4.28 Results from clustering evaluation - Mention CEAF metric.

4.29 Features computed from information available within the document.

4.30 Features computed from DBpedia.

4.31 Size metrics for the Reuters-128 entity salience dataset.

4.32 Results for different learning algorithms. † - learning the model for SVM on the NYT corpus takes more than 24 hours to process.

4.33 Evaluation results for different baseline methods for the “most salient” class.

4.34 Evaluation results for our method and the NYT method [1].

4.35 Evaluation results for different features.

4.36 Results for different portions of incorrectly linked entities.

4.37 Size metrics for the Tweets and News datasets.

4.38 Interannotator agreement for the core fields. The MSM dataset already contained a coarse grained type assignment, which was not reannotated.

4.39 Evaluation results for the EntityClassifier.eu NER system on the Tweets benchmark dataset.

4.40 Evaluation results for the Entityclassifier.eu NER system on the News benchmark dataset.

5.1 Top 5 resources with highest (left) and lowest RIC (right).

5.2 Top 5 most similar users with the user Alfredo (left) and the top 5 most relevant Web APIs (right).

5.3 Evaluation results for: Area Under the Curve (AUC), Normalized Discounted Cumulative Gain (NDCG), Mean Average Precision (MAP), Mean Reciprocal Rank (MRR).
5.4 Results from the evaluation of serendipity and diversity . . . . . . . . . . . . . 145
## Abbreviations

### Miscellaneous Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDF</td>
<td>Resource Description Framework</td>
</tr>
<tr>
<td>RDFS</td>
<td>Resource Description Framework Schema</td>
</tr>
<tr>
<td>SPARQL</td>
<td>SPARQL Protocol and RDF Query Language</td>
</tr>
<tr>
<td>OWL</td>
<td>Web Ontology Language</td>
</tr>
<tr>
<td>LOD</td>
<td>Linked Open Data</td>
</tr>
<tr>
<td>URI</td>
<td>Uniform Resource Identifier</td>
</tr>
<tr>
<td>IRI</td>
<td>Internationalized Resource Identifier</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>HTML</td>
<td>Hypertext Markup Language</td>
</tr>
<tr>
<td>NER</td>
<td>Named Entity Recognition</td>
</tr>
<tr>
<td>EL</td>
<td>Entity Linking</td>
</tr>
<tr>
<td>NIF</td>
<td>NLP Interchange Format</td>
</tr>
<tr>
<td>RS</td>
<td>Recommender Systems</td>
</tr>
<tr>
<td>CF</td>
<td>Collaborative Filtering</td>
</tr>
<tr>
<td>CBF</td>
<td>Content Based Filtering</td>
</tr>
<tr>
<td>SWS</td>
<td>Semantic Web Services</td>
</tr>
<tr>
<td>WSDL</td>
<td>Web Services Description Language</td>
</tr>
<tr>
<td>WSMO-Lite</td>
<td>Lightweight Semantic Descriptions for Services</td>
</tr>
<tr>
<td>WSMO</td>
<td>Web Service Modeling Ontology</td>
</tr>
<tr>
<td>OWL-S</td>
<td>Semantic Markup for Web Services</td>
</tr>
<tr>
<td>SAWSDL</td>
<td>Semantic Annotations for Web Service Description Language and XML Schema</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>FOAF</td>
<td>Friend of a Friend</td>
</tr>
<tr>
<td>JAPE</td>
<td>Java Annotation Patterns Engine</td>
</tr>
<tr>
<td>POS</td>
<td>Part-Of-Speech</td>
</tr>
<tr>
<td>CRF</td>
<td>Conditional Random Fields</td>
</tr>
</tbody>
</table>
Contributions and Algorithms related Abbreviations

RIC    Resource Information Content
AUC    Area Under the Curve
MAP    Mean Average Precision
NDCG   Normalized Discounted Cumulative Gain
MRR    Mean Reciprocal Rank
MFS    Most Frequent Sense
ESA    Explicit Semantic Analysis
MSM    Making Sense of Microposts
ESS    Exploratory Search Systems
TAC    Text Analysis Conference
PW     ProgrammableWeb
TF-IDF Term Frequency-Inverse Document Frequency
VSM    Vectors Space Model
SFI    Surface Form Index
LB     Basic Lucene index
LSD    Lucene Skip Disambiguation
SFS    Surface Form Similarity
ECC    Entity Co-occurrence
ASR    Automated Speech Recognition
Chapter 1

Introduction

The World Wide Web (WWW), since the inception, has the ultimate goal to enable efficient publishing and information sharing. It provides means for publishing documents and other resources as part of the global information space. Published documents are interlinked by hypertext links which enable users to navigate and discover new documents and information. The availability of such information space and the capability of the human being to discover and process the information is highly important for the success, future and existence of the human being.

Despite the benefits that the Web provides, until recently the principles that support the Web of Documents have not been applied on data. The data has been published in different formats (e.g. HTML, XML, CSV) without the capability to describe entities, express facts about entities and connect entities with typed links where the semantics of the links are formally defined.

Motivated by these issues, in the recent years, Linked Data [2] has emerged as a mechanism which employs the Web as a backbone for publishing, sharing and integration of data, information and knowledge. The Linked Data best practices have already attracted many communities from many domains, such as life sciences, government, geography and linguistics. These communities have published 1,146 datasets (as of January 2017\(^1\)), which is an overall growth of 388% compared to only 294 datasets published in September 2011. In spite of this increase, several problems are starting to surface. First, although there are domains which are well covered in the Linked Open Data (LOD) cloud, there are domains which are marginally or not covered at all. A quintessential domain is the Web services domain, which lack semantic descriptions published as Linked Data. Second, while Linked Data aims at publishing integrated knowledge, still huge amounts of valuable knowledge is hidden in unstructured format in text documents. Thus, knowledge from unstructured sources should be extracted and integrated with the LOD cloud. And third, due to the huge and diverse amount of information published as Linked Data, the actual retrieval of information from the LOD cloud demands significant amount of effort. The standard Linked Data access mechanisms such as SPARQL and URI dereferencing require

\(^1\)http://lod-cloud.net
1. Introduction

background knowledge on the technologies and data model, and have query expressivity limitations.

In this thesis, we investigate following research areas:

- knowledge acquisition and semantization for the Web services domain,
- knowledge extraction and integration using salient named entities, and
- personalized retrieval of Linked Data.

1.1 Motivation Scenarios

In this section, we present two scenarios to motivate our work. The first scenario presents the motivations for our work on knowledge acquisition, semantization and personalized retrieval of Linked Data. The second scenario illustrates the synergy between knowledge extraction and personalization in the context of Linked Data.

Scenario 1: Developing Mashup Applications. Bob is an application developer and he wants to improve the tourists’ experience in the Czech Republic by creating a Travel Mashup. His initial idea is to allow visitors to find and locate accommodation and interesting events. As a starting point, he uses HTML based search interface offered by ProgrammableWeb, the largest mashup and API directory, to search for similar mashups and improve his overall idea. Unfortunately, there are 593 travel mashups, thus he only focuses on the latest five travel mashups, leaving a high chance to miss some interesting mashups. Next, Bob starts to identify relevant APIs for his mashup by listing the “hotels”, “events” and “mapping” API categories. He wants to use a hotel API and enable users find accommodation and read user reviews for the hotels. The event API will be used to suggest events for the user location and the given holidays period. Finally, a mapping API will be used to locate and display on a map the discovered events and hotels. The hotels API category contains 102 APIs, the events category 269 APIs and the mappings category 957 APIs.

When selecting appropriate APIs, Bob is faced with several problems. First, reviewing all APIs from each category is time-consuming. Second, Bob is limited with the search query expressivity. For example, he is unable to automatically search for APIs that have been previously used in mashups with his already selected APIs. And finally, since the information is only provided in HTML, he is unable to automate the API search and discovery step.

Opportunities: formally describe and enhance the information from ProgrammableWeb with semantics and publish the information as Linked Data. This will enable users express their search requirements via SPARQL, allow other users to access the information and benefit from additional knowledge provided as Linked Data. We also see an opportunity in personalization of the API search. Since Bob regularly develops mashups, his past history and history information of the other users can be used to personalize the search results and provide relevant API recommendations. This will reduce the effort spent on searching for
relevant APIs, thus the mashup developers can focus more on the API integration rather on the API selection phase.

In order to address the aforementioned challenges, we have developed the largest dataset with semantic Web API descriptions (see Chapter 3). The collected data has been enhanced with semantics and published as Linked Data. This enables users to express their requirements by developing custom queries and executing them at a dedicated SPARQL endpoint which hosts the dataset. With regards to the personalized API recommendations, we have developed a method for personalized retrieval of Linked Data and validated it on the Web services domain (see Chapter 5).

**Scenario 2: Knowledge Extraction from Video Subtitles.** Alice is a farmer from a rural district in Germany. At her field she primarily grows asparagus. Since it is the start of the asparagus season in Germany, she decides to watch the latest news on TV related to asparagus. The videos she watches are accompanied with subtitles. In the news, she learns that this season the asparagus price will vary between different companies. Several companies are also mentioned in the subtitles. Thus, she decides to find out more background information about these companies on the internet. Also, she wants to cook dinner using asparagus, thus she searches for asparagus recipes.

When searching for relevant information on the companies and asparagus recipes, Alice is faced with several problems. First, searching for relevant information is time consuming. Second, she lacks background knowledge on the mentioned entities (companies, ingredients, recipes) and their importance in the video subtitles or in a Web document where these entities are mentioned.

**Opportunities:** Knowledge from the subtitles and other text documents can be extracted by utilization of NLP techniques, such as Named Entity Recognition (NER). NER can be used to extract information, such as the mentions of the companies and the asparagus plant from the video subtitles. Furthermore, these identified concepts (named entities) via entity linking can be linked with Linked Data datasets (e.g. DBpedia), which can help to acquire the required information (i.e. background information on the companies and asparagus recipes). For example, a mention of the “Alnatura” company could be linked to the DBpedia resource (http://dbpedia.org/resource/Alnatura) and then, relevant information about the company, such as its revenue, the year it was founded or the homepage of the company, can be retrieved. Similarly, the mention of the “asparagus” vegetable could be linked to the DBpedia resource (http://dbpedia.org/resource/Asparagus) and then, utilize the available information in DBpedia to collect list of recipes which include asparagus as an ingredient. For example, asparagus occurs as an ingredient in the Hochzeitssuppe soup and in the Crab Louie salad. Furthermore, many companies or vegetables could be mentioned in the video subtitle. Identifying a subset of salient entities that play an important role in the story that the document describes is important so that Alice can focus only on the most important entities (i.e. companies and vegetables). We also see an opportunity in personalization of the search query results. Since Alice has already watched

\(^2\)http://dbpedia.org/resource/Hochzeitssuppe

\(^3\)http://dbpedia.org/resource/Crab_Louie
1. Introduction

a video related to asparagus and some related companies, this information can be used as an implicit knowledge to provide relevant recommendations. As a consequence, this will enable Alice to retrieve more relevant information with less effort.

In our work, we address the aforementioned challenges related to knowledge extraction via exploitation of NER and EL techniques to extract relevant knowledge from unstructured content, i.e. textual documents, video subtitles and transcripts, and further integrate (i.e. establish links) this knowledge with existing information in the LOD cloud (see Section 4.1). We also address the challenge on identification of most important entities (i.e. salient entities) in the document and develop method for learning entity salience (see Section 4.2). Finally, in our work, we also address the challenge on personalized retrieval of knowledge in the context of Linked Data (see Chapter 5).

1.2 Problem Statement

In the last decade, Linked Data emerged as an efficient mechanisms for publishing, integrating and sharing of data, information and knowledge [2]. The efforts put by the Semantic Web community have gave birth of many datasets from various domains. Nevertheless, in spite of the noticeable increase in popularity, few of the key prerequisites necessary for Linked Data to keep its momentum are as follows: i) different domains in the LOD cloud should be well covered, ii) structured Linked Data knowledge has to be integrated with knowledge hidden in unstructured format (i.e. text documents), and iii) users should be able to easily retrieve Linked Data.

When creating and publishing Linked Data, data publishers face with several challenges. First, for particular domains, the data is not available in a machine-readable format; it can not be directly processed and converted into RDF. And second, for some domains, such as the Web services domain, there are no ontologies and vocabularies available to capture and model the available information, or the existing models are very complex and do not meet the requirements. The Web services domain is a quintessential domain which is not covered in the LOD cloud, and at the same time, it is affected by the two above-mentioned problems. In particular, vast majority of Web service descriptions are available only in non machine-readable format (in HTML), and existing semantic Web service models only partially capture this available information. The problem addressed in this dissertation with respect to the knowledge acquisition and semantization in the context of the Web services is on acquisition of data, enhancement of the data with semantics and publishing it as Linked Data.

While Linked Data aims at publishing and integrating structured knowledge, vast amount of knowledge is still hidden in an unstructured format and left unintegrated with the LOD cloud. Extracting and integrating knowledge from text documents with a structured knowledge from the LOD cloud has already shown positive impact on the performance of information retrieval [3] or as key enabler for semantic search [4]. Thus, there is need to extract the hidden knowledge found in text documents and integrate it with the existing Linked Data. Moreover, only the most important pieces of information should be extracted
and integrated. Named Entity Recognition and Entity Linking are the two widely exploited techniques for knowledge extraction and integration, which we investigate in this thesis. The problem addressed in this dissertation with respect to the knowledge extraction and integration using named entities is on efficient extraction of important knowledge in terms of named entities from text documents and its integration with the LOD cloud.

Published and integrated knowledge available as Linked Data should be also possible to easily retrieve and consume. Nevertheless, due to the huge and diverse amount of information, the actual retrieval of an information from the LOD cloud still demands significant amount of effort. The standard access mechanisms, SPARQL and URI dereferencing, have several limitations. First, they require background knowledge on the schema and the data model. And second, they have query expressivity limitations. In other words, in order to retrieve information, users can either write, usually complex, SPARQL queries, or dereference an initial set of seed URIs and iteratively follow the additional links, explore the graph and find relevant information. This, however, requires significant amount of effort from the users. Linked Data Recommenders aim at solving these problems by identifying, ranking and providing personalized resource recommendations (i.e. resources of interest for the user). The problem addressed in this dissertation with respect to the personalized knowledge retrieval in the context of Linked Data is on efficient recommendation of Linked Data resources of interest for the user.

The ultimate goal of the thesis is to address the above-mentioned problems related to Linked Data based Knowledge Provisioning, and in particular, i) knowledge acquisition and semantization for the Web services domain, ii) knowledge extraction and integration using salient named entities, and iii) personalized retrieval of knowledge in the context of Linked Data. Figure 1.1 depicts the thesis overarching process on “Linked Data based Knowledge Provisioning” and highlights the key areas of this dissertation thesis. The overarching process spans three activities:

- Activity 1. Knowledge Acquisition and Semantization: deals with acquisition of domain-specific knowledge and its semantization. In this thesis, we focus on acqui-
1. Introduction

The work presented in the dissertation has been done as part of several different European and national projects from different domains, with different requirements and goals. Nevertheless, the individual contributions are complementary and encompass an overarching process of knowledge provisioning in the context of Linked Data. The “Knowledge Acquisition and Semantization” activity contributes by acquisition and provisioning of domain-specific structured knowledge enhanced with semantics. The main result is a dataset with semantic Web service descriptions published as Linked Data (see Chapter 3 on Acquisition and Semantization of Web API Descriptions). Next, the “Knowledge Extraction and Integration” activity extracts knowledge from unstructured sources and integrates it with Linked Data. The main result from the second activity is an NER system supported with several methods for identification of salient named entities (see Chapter 4 on Knowledge Extraction and Integration with Salient Linked Entities). In the thesis, the developed NER system is applied on the results from the first activity and particular information from the Web service descriptions has been linked with Linked Data resources (see Section 3.5). Finally, the “Personalized Knowledge Retrieval” activity exploits the knowledge generated by the first two activities and enables personalized retrieval of Linked Data. We develop a method for personalized retrieval of Linked Data (see Chapter 5) which is validated and applied on the results from the first and second activity, the dataset with semantic Web service descriptions.

Each activity from the thesis addresses specific challenges. In this section, we define our research questions (RQ) and their associated hypotheses (H) which frame the thesis with respect to each activity. We start by defining the research questions which frame our research so that we can provide answers and evidence for the hypotheses. We define set of research questions and hypotheses for each activity from the thesis overarching process.

1.2.1 Knowledge Acquisition and Semantization

While the LOD cloud has significantly increased over the past decade, there are domains, such as the Web services domain, which are not covered in the LOD cloud. There are two main challenges when publishing Linked Data for the Web services domain. First, great majority of the Web services descriptions are not available in a machine-readable format. And second, existing ontologies and vocabularies are too complex or they do not completely capture the available information. As a consequence, researchers and practitioners are
unable to develop efficient Web service query mechanisms, existing Web service descriptions are not contextualized, integrated and linked, and analysts can not properly analyze the Web service ecosystem and trends. Our work on knowledge acquisition and semantization of Web service descriptions is guided by the following set of research questions on which we provide answers:

- **RQ1.1:** “What are the benefits of a dataset with semantic Web service descriptions?”
- **RQ1.2:** “To what extent do semantics improve the accuracy of the process of Web service retrieval?”
- **RQ1.3:** “How can we improve the efficiency of modeling relevant Web service information?”
- **RQ1.4:** “To what extent do different types of Web service users find a dataset with semantic Web service descriptions useful?”

In our work, we collect Web services related information and develop an integrated semantic model. We implement this as a light-weight ontology which captures all available information. In order to support our work on the creation of the dataset, we develop several use cases and illustrate the benefits of the dataset. We also develop a Linked Data based recommendation method which confirms the benefit of using semantics in the process of Web service retrieval. Further, we execute a survey on the usefulness of the dataset and confirm its potential.

The main results of our work are: a dataset with semantic Web API descriptions, a light-weight ontology for modelling relevant Web API information, and a survey on the usefulness of the dataset. Moreover, in relation to the research questions listed above, we provide evidence for the following hypotheses:

- **H1.1:** “A dataset with semantic Web Service descriptions enables *advanced analysis of the service ecosystem that was not possible before.*”
  We identify the advantages of a dataset with semantic Web service descriptions in analysis of the service ecosystem (see Section 3.7.1). In particular, it enables comparison of the popularity and consumption of different Web services over time, identification of latest trends (e.g. protocol or data formats), or investigate the API popularity across different domains. Analysis of the services ecosystem have not be able to execute before without the availability of a dataset with semantic Web service descriptions.

- **H1.2** “A dataset with semantic Web service descriptions enables *more accurate retrieval of Web services.*”
  We have executed an experiment and provide evidence that our method which considers *semantics* provides more accurate results compared to the traditional non-semantic based recommendation mechanisms (see Section 5.3.3, Experiment 1). The results show an accuracy increase (Area Under the Curve measure) of 87%, 7%, 49% and 34% for four different non-semantic based recommendation mechanisms.
1. Introduction

○ H1.3: “An integrated light-weight ontology enables to efficiently capture all relevant Web service information that was not possible before.”

We conducted a literature review of the existing semantic Web service models (see Section 2.2.1.1) and the available Web service data sources (see Section 2.2.1.2), and based on the findings from the review, we have developed an integrated light-weight ontology which captures all relevant Web services information (see Section 3.3). On a real-world data from the Web services domain (see Section 3.4) we show the capabilities of the developed ontology.

○ H1.4: “Vast majority of the Web service users find a dataset with semantic Web descriptions useful.”

In order to evaluate the usefulness of the dataset, we have executed a survey targeting Web service consumers and providers (see Section 3.7.2). According to the results, 17% of the consumers find the dataset very useful, 48% useful and 35% somewhat useful. From the Web service providers, 5% find the dataset very useful, 48% useful, 26% somewhat useful, 16% little useful, and only 5% not useful at all.

1.2.2 Knowledge Extraction and Integration

While the previous activity deals with knowledge acquisition and semantization, it does not solve the problem of the knowledge extraction from unstructured sources (e.g. text documents) and its integration with the LOD cloud. The fact is that vast amount of knowledge is still hidden in an unstructured format and left unintegrated with the LOD cloud. Named entity recognition (NER) and entity linking (EL) are two widely exploited techniques for knowledge extraction and its integration with the LOD cloud. In the past decade, number of NER and EL systems have been developed, however, they heavily rely on training data and they do not evaluate the importance of the entities in relation to the document. In our work, we address these two problems and develop methods for unsupervised entity spotting and classification, as well as, a method for identification of salient entities. Our work on knowledge extraction and integration using salient named entities is guided by the following set of research questions on which we provide answers:

○ RQ2.1: “How does the quality of our NER system compare to other systems for different datasets and for different type of focus queries?”

○ RQ2.2: “How accurate results gives our method for identification of salient named entities compared to other similar methods?”

○ RQ2.3: “What is the impact of individual and combined local and global set of features on the performance of learning entity salience?”

○ RQ2.4: “How does the quality of the entity links influence the performance of learning entity salience?”

In our work, we developed an NER system named Entityclassifier.eu, which is supported with several methods for entity spotting, linking, classification, as well as, identification
of salient entities. The developed entity spotting method is supported with a lexico-syntactic patterns and it does not require any training data, while the entity classification is performed at query time by mining the types in real-time from Wikipedia. The NER system also evaluates the salience (importance) of the recognized entities. We developed a supervised method for identification of salient entities. The method considers local information about entities derived from the documents content, and global information about the entities from an external knowledge graph (i.e. DBpedia). In relation to the research questions listed above, we provide evidence for the following hypotheses:

○ H2.1: “Our NER system gives more accurate results compared to other related systems.”
We executed several experiments in order to evaluate the performance of our system and compare it with other related systems. On an experiment using the Czech Traveler dataset (see Section 4.1.5.1) our system outperformed three state of the art systems. Our system achieved 0.61 F1 for entity spotting (second best by AlchemyAPI 0.58 F1), 0.66 F1 for entity classification (second best by OpenCalais and AlchemyAPI 0.45 F1) and 0.67 F1 for entity linking (DBpedia Spotlight achieved 0.43 F1). It worth mentioning that our methods for entity spotting and classification are fully unsupervised in comparison to the supervised methods implemented as part of the other NER systems (e.g. DBpedia Spotlight). Also, we show that our most-frequent-sense based linking approach outperforms the context based linking approach, which is used by DBpedia Spotlight. We also evaluated the performance of our system for different focus queries (PER, ORG, GPE) and different document collections (newswire, web documents, discussion fora documents) at the TAC 2013 challenge (see Section 4.1.5.3) and the results show that our most-frequent-sense based linking approach achieved best results for the GPE focus queries (0.677 F1) and for the discussion fora documents (0.539 F1).

○ H2.2: “Our method for learning entity salience gives more precise results than the state of the art method proposed by Dunietz and Gillick [1].”
We executed an experiment and evaluated the performance of our method in comparison to the most related method developed by Dunietz and Gillick [1]. According to the results, our method achieved better precision (0.611) compared to the method developed by Dunietz and Gillick [1] (0.605) (see Section 4.2.4, Experiment 2). The key difference between our method and the other method [1] is that when computing the global features we use the whole external knowledge entity graph (i.e. DBpedia), while in the other method, the graph feature is computed only from graph consisting of entities occurring in the document. Moreover, we consider several graph metrics, such as PageRank, HITS, in-degree and out-degree, while in their work they only consider PageRank.

○ H2.3: “A combined set of local and global features gives better accuracy than each set used individually.”
We executed an experimental evaluation and analyzed the impact of the individual
features on the performance of learning entity salience (see Section 4.2.4, Experiment 3). The results show that a model which considers both, local and global features, achieves better results (0.607 F1) than the model which considers the local (0.592 F1) or the global features (0.489 F1) only.

- H2.4: “Incorrect links have low impact on the performance of learning entity salience.” We evaluated the impact of the quality of the entity linking on the performance of learning entity salience (see Section 4.2.4, Experiment 3). The results show that 10% of incorrectly linked entities results in 2.47% decrease, while 20-25% of incorrectly linked entities result in 4.12% decrease of the accuracy of learning entity salience. It can be also observed, that also with 50% of incorrectly linked entities, the learning accuracy still shows promising results (0.553 F1).

1.2.3 Personalized Knowledge Retrieval

With the increasing number of integrated Linked Data datasets, the actual retrieval of Linked Data information is becoming extremely difficult. Manual assessment of the appropriateness of the information for specific users is inefficient and it demands significant amount of effort. The standard access mechanisms, SPARQL and URI dereferencing, require background knowledge on the schema and have expressivity limitations. In order to overcome these issues, several Linked Data recommendation methods have been developed. However, these methods are usually developed for a particular domain or dataset, they require manual pre-processing of the dataset, or they exploit only the instance data, while the schema is rarely considered. Our work on Personalized Knowledge Retrieval is guided by the following set of research questions on which we provide answers:

- RQ3.1: “How does the quality of our Linked Data recommendation method compare to the traditional personalised and non-personalised methods on a dataset from the Web services domain?”
- RQ3.2: “To what extent the resource informativeness influences the accuracy, serendipity and diversity of the Linked Data recommendations?”
- RQ3.3: “What is the impact of the serendipity and diversity on the accuracy of the recommendations?”

In our work, we developed a novel method for personalized retrieval of Linked Data resources of interest for the users. The method takes into account the commonalities, the informativeness and the connectiviteness of the resources. Moreover, the method is not domain or dataset specific and does not require any manual pre-processing of the datasets; it exploits RDF datasets in their original form. We evaluate our method and show that the method outperforms the traditional personalised and non-personalised methods in terms of accuracy, serendipity and diversity. We also evaluate and provide evidence on the trade-off between serendipity, diversity and accuracy of the recommendations. In relation to the research questions listed above, we provide evidence for the following hypotheses:
1.3. Related Work and Previous Results

- H3.1: “Our method, which takes into account the commonalities, the informativeness and the connectiviteness of the resources, outperforms the other personalized and non-personalized methods in terms of accuracy, diversity and serendipity.”

We executed an experimental evaluation and compared our method to the traditional personalized and non-personalized methods (see Section 5.3.3). The results show that our method, which considers the dataset semantics, the commonalities, informativeness and the connectiviteness of the resources, outperforms the other personalized and non-personalized methods in terms of accuracy (0.95093 F1\(^4\)), serendipity (3.42444) and diversity (0.83114).

- H3.2: “Resource informativeness improves the accuracy but decreases the serendipity and diversity of Linked Data recommendations.”

We analyzed the impact of the resource informativeness on the accuracy, serendipity and diversity of the recommendations (see Section 5.3.3, Experiment 2). The results show that the resource informativeness has positive impact on the accuracy (16.98% for the ACU measure), and negative impact on the diversity (10.85%) and serendipity (6.02%).

- H3.3: “A result set with more serendipitous and more diverse gives less accurate recommendations.”

We analyzed the trade-off between serendipity, diversity and accuracy of the recommendations. The result show that serendipity and diversity have negative impact on the accuracy of the recommendations. According to our observations, the optimal values for serendipity at 3.2 are precision at 0.12 and recall at 0.77. Optimal values for diversity at 0.825 are precision 0.13 and recall at 0.763.

1.3 Related Work and Previous Results

In this thesis, we investigate three research areas: i) acquisition and semantization of Web service descriptions, ii) knowledge extraction and integration with salient named entities, and iii) personalized retrieval of Linked Data. This section briefly introduces the related work, previous results and open challenges in these three areas. A comprehensive review of the related work and previous results in these areas can be found in Section 2.2.

Acquisition and Semantization of Web Service Descriptions. In the past years, Semantic Web community has been primarily focused on developing technical infrastructure and technologies for efficient Linked Data acquisition, publishing and interlinking. These efforts gave birth of 1,146 datasets (as of January 2017\(^5\)) covering different domains, such as life sciences, government, geography and linguistics. Nevertheless, there are specific domains which are not well covered or not covered at all. A particular of a domain with no coverage in the LOD cloud is the Web services domain. In our work, we focus on knowledge acquisition, semantization and Linked Data publishing for the Web services domain. There

---

\(^4\)A score achieved for the Area Under the Curve (ACU) measure.

\(^5\)http://lod-cloud.net
are two key requirements when publishing Linked Data for a specific domain: i) availability of data in machine-readable format and ii) availability of ontologies and vocabularies for modelling the data.

With regards to the available ontologies and vocabularies, in the last decade, several semantic Web service models have been proposed. OWL-S\(^6\)\(^5\) was the first major semantic Web service model with the goal to enable automated Web service provisioning. In addition, several other semantic Web service models, such as WSMO \(^6\)\(^6\)\(^5\), SAWSDL\(^7\), WSMO-lite \(^7\), hRESTS \(^8\), SADI \(^9\) and Hydra\(^8\), have been also proposed. Nevertheless, great majority of the semantic Web service models have failed to gain a significant uptake on the Web. The prime reason for the minimal uptake of the semantic Web service models is the high diversity of incompatible and complex solutions \(^\)\(^10\) which require significant effort from the users \(^11\). Furthermore, some of them address Web service models (WS-DL/SOAP) which are nowadays not prevalent on the Web \(^12\). Another problem with the existing solutions is that they do not completely capture the Web API information such as the functional, non-functional, technical, provenance and temporal information. And finally, existing solutions do not, or just partially take into account services and the users’ social context in the service directories, such as the relationships among the users, services and service providers, the way they collaborate and communities they create.

Despite the minimal uptake of the semantic Web service models, in the last few years, Web APIs enjoy significant increase in popularity. However, most of the Web APIs descriptions are nowadays provided in a non machine-readable format, in HTML. There are several initiatives which propose machine-readable formats for description of Web APIs, such as OpenAPI\(^9\) and APIs.json\(^10\), however, the number of published Web API descriptions with these formats is still very low. On contrary, ProgrammableWeb\(^11\), the largest mashup and API repository, lists over 17,000 APIs (as of July 2017), however, the API descriptions are provided in HTML. Similarly, API For That\(^12\) and Exicon\(^13\) repositories, also provides API descriptions. Nevertheless, the Web API descriptions are only provided in non human-readable format (i.e. HTML). In summary, there is no repository which provides Web API descriptions enhanced with semantics.

Overall, existing problems can be summarized as follows:

- ad-hoc API description solutions without semantics,
- lack of API descriptions in a machine-readable format,
- API descriptions with limited contextual information such as information on applications that consume the API, developers that used the API, or the time when the

\(^6\)https://www.w3.org/Submission/DWL-S/
\(^7\)http://www.w3.org/TR/sawSDL/
\(^8\)http://www.hydra-cg.com/spec/latest/core/
\(^9\)https://www.openapis.org/
\(^10\)http://apisjson.org/
\(^11\)https://www.programmableweb.com/category/all/apis
\(^12\)http://www.apiforthat.com/
\(^13\)https://app.exiconglobal.com/api-dir/
1.3. Related Work and Previous Results

API was developed/modified/used, and

- complex semantic web service models with partial Web service modelling capabilities.

All these problems make the discovery, sharing, integration, and assessment of their quality and consumption problematic. The lack of shareability and reuse are also reflected in higher maintenance and development costs of the Web APIs. An in-depth review of the related work and previous results related to semantic Web service descriptions can be found in Section 2.2.1.

Knowledge Extraction and Integration with Named Entities. While Linked Data aims at publishing integrated knowledge, yet huge amount of valuable information on the Web is hidden in text. In the recent years, the Named Entity Recognition (NER) and Entity Linking (EL) have been widely exploited as techniques for knowledge extraction from unstructured data and its integration with Linked Data. As a results, several NER and EL systems, such as DBpedia Spotlight [13], AIDA [14], NERD [15], Babelfy [16], FOX [17], OpenCalais\(^\text{14}\) and StandfordNER\(^\text{15}\), have been developed. Nevertheless, most of the NER systems require training data for the entity spotting and classification task [13, 16, 14].

Another big issue with the existing NER system is that they do not evaluate the importance (salience) of the entities in the document. AlchemyAPI and OpenCalais provide relevance scores for the entities, however, in our work we are interested in evaluation of entity salience, i.e. how important is the entity in the story that the document describes, which is different than the notion of “entity relevance”. Moreover, there is lack of a complete, publicly available and evaluated by human entity salience dataset. The two existing datasets, Microsoft Document Aboutness (MDA) [18] and the New York Times (NYT) [1], do not provide the documents due to copyright restrictions, the entity annotations have been generated automatically using a proprietary NER system and as a consequence contain incorrect annotations, and more importantly, the salience annotations have been also generated automatically and have not been manually checked.

Overall, two of the key main issues with the existing NER systems are as follows:

- they require training data for entity recognition and classification, and
- they do not assess entity salience in the document, and also, there is a lack of a dataset with entity salience annotations which is complete, publicly available and evaluated by human.

An in-depth review of the related work and previous results related to NER and EL can be found in Section 2.2.2.

Personlized Retrieval of Linked Data. Due to the huge and diverse amount of information published as Linked Data, the actual retrieval of an information from the LOD cloud still demands significant amount of effort. Linked Data Recommenders aim at solving these problem by identifying, ranking and recommending resources of interest for the

\(^\text{14}\)http://www.opencalais.com/
\(^\text{15}\)https://nlp.stanford.edu/software/CRF-NER.shtml
1. Introduction

user. In the recent years, several Linked Data recommendation approaches have been proposed. These approaches primarily adopt traditional recommendation techniques on Linked Data, or utilize graph-based techniques to analyze the graph and generate recommendations. Nevertheless, the developed methods have been designed and validated for a particular domain. While there are few methods [19, 20] which can be applied on multi-domain datasets, most of the methods usually require human expertise for the domain of interest [21, 22, 23, 24, 25, 26, 27, 28, 29], or require manual pre-processing of the domain dataset [30].

Another issue with the existing solutions is the dataset adaptation. Vast majority of the existing methods are exclusively designed for a specific dataset and their adoption is impossible or require significant human effort. The problem with dataset adaptation has been addressed by the RecSPARQL method [26], which enables development of customized recommenders for arbitrary RDF graphs. The Linked Data dynamics is another important aspect which is partially considered by the existing solutions. While most Linked Data recommenders process the datasets offline, DiscoveryHub [19] and RecSPARQL [26] support on-the-fly generation of recommendations based on fresh data. However, DiscoveryHub is dataset specific, exclusively developed for DBpedia, considers only instance data and exploits only a small portion of the available information in DBpedia. RecSPARQL requires human expertise to configure the method and select the features to be used when recommending resources of interest. Furthermore, both, instance data and the ontology schema, can provide valuable information for the Linked Data recommenders. However, vast majority of Linked Data recommenders only consider the instance data, while the schema is considered only in [31, 23, 24, 26].

Overall, existing problems can be summarized as follows:

- domain and dataset specific adaptation of the existing methods is difficult or not possible at all,
- real-time and fresh recommendations are rarely provided, and
- the available Linked Data information is not fully exploited.

An in-depth review of the related work and previous results related to Linked Data recommenders can be found in Section 2.2.3.

1.4 Contributions of the Thesis

The thesis provides contributions to the process of Linked Data based knowledge provisioning. In particular, the thesis provides contributions which are outcomes from the three activities of the overarching process i) knowledge acquisition and semantization, with focus on the Web Services domain; ii) knowledge extraction and integration, with focus on extraction of salient named entities; and iii) personalized knowledge retrieval, with focus on personalized retrieval of Linked Data.

The main contributions of the thesis related to the knowledge acquisition and semantization with focus on the Web Services domain, further detailed in Chapter 3, are threefold:
1.4. Contributions of the Thesis

- **CT1.1**: A Linked Data dataset with semantic Web API descriptions, largest of its kind. The datasets enables more advanced analysis of the Web service ecosystem that was not possible before (see Section 3.7.1), such as comparison of the popularity and consumption of different Web services over time, identification of latest trends (e.g. protocol or data formats), or investigation the API popularity across different domains. We also show that a Web service recommendation methods can benefit from the semantics and provide more accurate recommendation compared to the traditional non-semantic based recommendation methods (see Section 5.3.3).

- **CT1.2**: A light-weight ontology for modelling relevant Web API information, which enables efficiently to capture all relevant Web service information that was not possible before with the existing semantic models (see Section 3.3), and

- **CT1.3**: A survey on the usefulness of the dataset, which provides an evidence and ascertains the added value and the degree of achievement (see Section 3.7.2). We show that vast majority of the Web service users find a dataset with semantic Web descriptions useful.

The main contributions of the thesis related to the Knowledge Extraction and Integration activity with focus on extraction of salient named entities, further detailed in Chapter 4, are sixfold:

- **CT2.1**: An open-source NER system named Entityclassifier.eu and a set of methods for named entity spotting, linking and classification (see Section 4.1), which outperforms other related NER systems in terms of accuracy. These evaluations have been executed on the Czech Traveler dataset (see Section 4.1.5.1).

- **CT2.2**: An evidence on the accuracy of each developed method under different conditions (datasets, domains, focus queries and languages). We show that the most-frequent-sense approach gives more accurate results for Person focused queries compared to Organization and Geopolitical queries. We also evaluate the performance for different document collections and show that our system gives best results for the discussion fora documents than for the newswire or web documents. These experiments have been executed as part of our participation at the TAC 2013 (see Section 4.1.5.3) and TAC 2014 (see Section 4.1.5.4) challenges.

- **CT2.3**: A method for learning entity salience based on local and global set of features, which gives more precise results when compared to the most related state of the art method [1] (see Section 4.2.4).

- **CT2.4**: A crowdsourced dataset with entity salience annotations. Due to the lack of a publicly available, complete and manually checked dataset with entity salience annotations, we have developed such dataset via crowdsourcing (see Section 4.2.3).

- **CT2.5**: An evidence about the impact of the individual features on the performance of learning entity salience. We show that a combined set of local and global features gives better accuracy than each set used individually. We also evaluate the impact of each individual feature (see Section 4.2.4).
1. Introduction

- CT2.6: An evidence about the impact of the incorrect links on the performance of learning entity salience. We show that incorrect links have low impact on the performance of learning entity salience (see Section 4.2.4).

Finally, the main contributions of the thesis related to the Personalized Knowledge Retrieval in the context of Linked Data, further detailed in Chapter 5, are threefold:

- CT3.1: A method for personalized retrieval of Linked Data, which takes into account the commonalities, the informativeness and the connectiviteness of the resources. The method (see Section 5.2) considers both, instance data and ontology schema, it is dataset and domain independent and it outperforms the traditional personalised and non-personalised methods in terms of accuracy, serendipity and diversity (see Section 5.3.3).

- CT3.2: An evidence about the impact of the resource informativeness on the accuracy, serendipity and diversity of recommendations. We show that resource informativeness improves the accuracy but decreases the serendipity and diversity of Linked Data recommendations (see Section 5.3.3).

- CT3.3: An evidence about the trade-off between accuracy, serendipity and diversity of the recommendations. We show that a result set with more serendipitous and more diverse gives less accurate recommendations (see Section 5.3.3).

1.5 Research Methodology

In order to address the research questions listed in Section 1.2, we build our research upon the experimental methodology [32], which is commonly considered in the Computer Science research to develop and evaluate a new solution for a given problem.

Our research begins with an analysis of the state-of-the-art and an extensive literature review. This covers concepts, approaches and methods from different fields and in particular the Semantic Web (i.e. Linked Data), Web Engineering (i.e. Web services research), Information Extraction (i.e. Named Entity Recognition and Linking) and Information Retrieval (i.e. Recommender Systems). We focus our research around Linked Data and we identify open challenges and problems related to Linked Data, Web services and Knowledge Extraction using named entities. We also utilize concepts, methods, algorithms and best practices within the fields of Collective Intelligence, Graph Theory, Natural Language Processing, Linguistics, Machine Learning and Crowdsourcing, which we exploit in the design and development of our methods and problem solutions.

Next, we develop a conceptual and theoretical design of our proposed solutions and their experimental evaluation. We design our prototypes with the goal to validate our research hypotheses in an iterative fashion, as proposed in [33], where experimentation is used as a feedback loop towards improvement of our solutions and meeting the expected quality.
1.5. Research Methodology

Follows implementation of the conceptual architecture, the algorithms and the methods from the design phase. The implementation serves as a proof-of-concept in order to demonstrate the feasibility of the proposed solution. In order to demonstrate the feasibility of the proposed solutions, we have provided following implementations: a Linked Data dataset with comprehensive information about Web APIs (see Section 3.4), an open-source system for recognition, linking and classification of salient entities (see Section 4.1), datasets for NER and entity salience (see Section 4.2.3) and a method for personalized recommendation of Linked Data resources applied on a real-world data from the Web services domain (see Section 5.3.2).

The evaluation is a crucial step in our research methodology. It includes a design and execution of experiments and/or surveys to ascertain the added value and degree of achievement with respect to the related work. The methods and algorithms are evaluated on a real-world data and validated on real-world scenarios. Where needed, evaluation datasets are being developed using crowdsourcing or crawling data acquisition techniques. In particular, we designed and executed following set of experiments and survey: a survey on the usefulness of the Linked Data dataset with Web API descriptions (see Section 3.7.2), an evaluation of the NER system at the TAC 2013 and TAC 2014 entity discovery and linking challenges (see Section 4.1.5.3 and 4.1.5.4), an evaluation on real-world data provided by LinkedTV project (see Section 4.1.5.2), an evaluation on the Czech Traveler dataset (see Section 4.1.5), a case study on concept detection in video transcripts (see Section 4.1.5.5), an evaluation of the method for learning entity salience on a crowdsourced dataset (see Section 4.2.4) and an evaluation of the method for personalized retrieval of Linked Data on data from the Web services domain (see Section 5.3). In the following chapters (Chapter 3, 4 and 5) we approach each set of questions, propose solution and execute an experimental evaluation in order to assess whether our solution match the expectations. In an iterative fashion, the results from the experiments are used to improve our solutions.

As a last step we disseminate our work by publishing and presentation of the results in workshops, conferences, journals and books. The venues are selected based on their reputation and relevance within the community dealing with the topic of the work. In particular, our work has been published at venues driven by the Semantic Web, Knowledge Engineering, Knowledge Discovery, Natural Language Processing and Machine Learning communities, such as ISWC, EKAW, SWJ, ECMLPKDD, LREC, etc. According to the level of maturity of the research, the results are presented in the appropriate format; preliminary results are presented in workshop papers [A.9, A.10, A.17]; via poster and demo papers [A.4, A.6] the results are demonstrated and feedback from the community is gathered; complete research results with proper evaluation are presented at conferences [A.2, A.3, A.5]; and their in-depth extended versions are published as journal papers [A.1] and book chapters [A.11, A.12]. Note that most of the publications related to our work on knowledge extraction and integration using named entities are based on the work of the author of this thesis in the European LinkedTV\(^{16}\) and LOD2\(^{17}\) projects. The results of our

---

16 https://www.linkedtv.eu/
17 http://lod2.eu/
work have been also described in several technical reports [A.13, A.14, A.15]. A complete list of publications is provided in the “publications” section of the thesis.

1.6 Structure of the Thesis

The thesis is organized into following six chapters:

Chapter 1. Introduction: Introduces the thesis topic, motivates the work, lists the research questions which guide our work, lists the core contributions of the thesis and presents the research methodology.

Chapter 2. Background and State-of-the-Art: Introduces the reader to the necessary theoretical background on Semantic Web, knowledge extraction and integration and recommender systems. It also surveys the current state-of-the-art in Semantic Web Services, Named Entity Recognition, Entity Linking and Linked Data Recommenders.

Chapter 3. Acquisition and Semantization of Web API Descriptions: Describes the first contribution of the thesis, the largest dataset with semantic Web API descriptions. We motivate the creation of the dataset, describe the process of creation and present the developed ontology for modeling relevant Web APIs information. We also assess the quality and the usefulness of the dataset and we present the results. Several selected use cases are also presented.

Chapter 4. Knowledge Extraction and Integration with Salient Linked Entities: Describes several developed approaches on general Named Entity Recognition, Entity Linking and identification of salient entities. Section 4.1 describes a Named Entity Recognition system, named Entityclassifier.eu, and its underlying methods for entity spotting, classification and linking. We present the results from the evaluation of the system on several entity recognition and linking challenges and datasets. Section 4.2 describes a method for identification of salient entities. We also describe a novel corpus with crowdsourced entity salience annotations which supports our method. Section 4.3 describes an evaluation framework for NER, its components and datasets.

Chapter 5. Personalized Retrieval of Linked Data: Describes a method exclusively developed for personalised recommendation of Linked Data resources. We describe its definitions, algorithms and a use case on the Linked Web APIs dataset, which is described in Chapter 3. We also describe several experiments we run to validate and evaluate the method.

Chapter 6. Conclusions and Future Work: Summarizes the results of our research, suggests possible future research directions and concludes the thesis.
Background and State-of-the-Art

This chapter introduces the reader with all necessary theoretical background and surveys the current state-of-the-art in the thesis topics. Section 2.1 summarizes the theories, techniques and fundamentals of the Semantic Web and Linked Data, Knowledge Extraction and Integration, general Recommender Systems and Linked Data Recommenders. Section 2.2 summarizes the previous results in the thesis topics. In particular, it surveys the most recent results and identifies the research gaps in the following fields: Semantic Web Services, Named Entity Recognition and Entity Linking, and Linked Data Recommenders.

2.1 Theoretical Background

There are three research lines which provide the conceptual underpinning for thesis.

- **Semantic Web and Linked Data**: the research conducted in this thesis is primarily focused on Semantic Web and Linked Data technologies. In particular, we focus on development of semantic models, methods for knowledge extraction and integration with Linked Data and methods for efficient retrieval of Linked Data.

- **Knowledge Extraction and Integration**: the thesis partially deals with extraction of important knowledge from unstructured content, formal representation of the extracted knowledge and its integration with existing knowledge sources.

- **Recommender Systems**: conducted research is reusing concepts from the recommender systems research and adopts them to Linked Data.

2.1.1 Semantic Web

The idea of adding “semantics” to the Web has been pursued by Tim Berners-Lee already in early 1990s. However, the term “Semantic Web” gained attention from the Web community in May 2001 when Tim Berners-Lee et al. published the seminal article entitled “The Semantic Web” [34]. In this article, the Semantic Web is defined as an extension of the
2. Background and State-of-the-Art

Web, in which information is given well-defined meaning with the goal to enable computers and humans to work in cooperation. Furthermore, the authors of the article put notion on the need for knowledge representation, reasoning over it and defining common meaning using ontologies, with the ultimate goal of enabling intelligent agents to collect content from diverse sources, process the information and exchange the results with other agents.

In the nutshell, the Semantic Web can be seen as an extension of the traditional Web through standards. The key foundations of the Semantic Web are data formats that can be used to model, share and process knowledge with unambiguous meaning.

The main goal here is to provide knowledge that can be understandable not only by humans, but also machines. In general, there are three topics which define the foundations for the Semantic Web [35]. The first topic is on building semantic models which aims at describing world concepts in abstract terms and thus enable easier understanding of the described system. The second topic is focused on computing with knowledge which deals with reasoning over encoded knowledge and deriving meaningful conclusions. And finally, the third topic is on efficient exchange of information among computer agents. The goal is to define best practices for publishing, sharing and interlinking structured information on a global scale.

The Semantic Web standards are being primarily developed by the World Wide Web Consortium (W3C)\(^1\). However, there are also other initiatives which deal with Semantic Web technologies. One of those initiatives is schema.org\(^2\), which is a collaborative community effort with the goal to develop, maintain and promote common schemas for structured data on the Web. It has been initially established by Google, Microsoft, and Yahoo.

2.1.1 RDF

The Resource Description Framework (RDF)\(^3\) is a formal data model for describing structured information. It is defined as framework for expressing information about resources. The foundation structure of RDF is a triple, which consist of a subject, a predicate and an object.

**Resource.** In the Semantic Web, a resource can be anything. For example, web documents, people, companies, physical objects, but also abstract concepts such as location addresses, can be considered as resources. Each resource is identified with an Internationalized Resource Identifier (IRI)\(^4\). These identifiers should be considered as unique on the global scale. A resource denoted with an IRI is also known as its referent. The benefit of using IRIs is that they can be dereferenced and a representation of the resource behind the IRI can be retrieved.

**Property.** Property is defined as a resource that can be thought of as a binary relation. Properties are also identified with IRIs and are used to describe resources. For example,

---

\(^1\)https://www.w3.org/

\(^2\)https://schema.org/

\(^3\)RDF 1.1 Concepts and Abstract Syntax W3C Recommendation: https://www.w3.org/TR/rdf11-concepts/

\(^4\)In RDF 1.0 identifiers for resources are URIs, while in RDF 1.1 are IRIs.
2.1. Theoretical Background

![RDF Graph Example](image)

Figure 2.1: A simple RDF graph example.

the property [http://xmlns.com/foaf/0.1/name](http://xmlns.com/foaf/0.1/name) (foaf:name) denotes a name for a thing. Figure 2.1 provides a simple example where the foaf:name property is used to indicate the name for the resource [http://example.org/#Bob](http://example.org/#Bob).

**RDF statement.** An RDF triple is also known as an RDF statement. RDF statements are used to describe properties for resources. Each statement consists of a subject, a predicate and an object where the first resource is the subject of the statement and the second resource is the object of the statement, while the relationship between these resources is known as a predicate or a property. The example from Figure 2.1 consists of six statements/triples, among which is a statement which gives the name of the person:

```
<http://example.org/#Bob> <http://xmlns.com/foaf/0.1/name> "Bob Dylan"
```

**RDF Document.** An RDF document is defined as a directed graph consisting of nodes linked by directed edges. An RDF document encodes an RDF graph in a concrete RDF syntax. Figure 2.1 shows an example of an RDF graph using the graph notation. The same graph can be represented in other RDF syntaxes, also known as serialization formats, such as Turtle[^5], RDF/XML[^6] or JSON-LD[^7]. In Listing 2.1 is shown the same graph from Figure 2.1 in the Turtle serialization format.

[^5]: https://www.w3.org/TR/turtle/
[^6]: https://www.w3.org/TR/rdf-syntax-grammar/
[^7]: https://www.w3.org/TR/json-ld/
2. Background and State-of-the-Art

In the computer science, an ontology describes a knowledge about a domain of interest, the core of which is a machine-processable specification with a formally defined meaning of the described concepts and relationships [35]. The ultimate goal of defining ontologies is to avoid conceptual and terminological confusion in software systems [36]. Using an ontology one can represent knowledge about a given domain including concepts, their properties, and relations among those concepts.

**RDF Schema** (short RDFS)

provides a data-modeling vocabulary for RDF data. It defines vocabulary with generic language constructs which can be used to semantically describe groups of related resources and the relationships between these resources. RDFS builds on top of RDF and extends the RDF vocabulary. Its expressivity is limited, but powerful and sufficient for many use cases. RDFS has been mostly exploited to describe classes of objects (rdfs:Class), hierarchy of classes (rdfs:subClassOf) and their properties (rdfs:subPropertyOf). Further, RDFS can be used to express the domain (rdfs:domain—source class) and the range (rdfs:target—target class) for the properties.

Although RDFS can be used to model and derive implicit knowledge, it provides limited level of expressivity which is required for representation of complex knowledge spaces. To solve this gap, the Web Ontology Language\(^9\) (OWL) has been defined. OWL is based on description logics and it allows logical reasoning on the knowledge and access to knowledge which is implicitly modeled. Since 2004, OWL is a W3C recommended standard for modeling ontologies.

---

\(^8\)https://www.w3.org/TR/rdf-schema/

\(^9\)https://www.w3.org/TR/owl-ref/
2.1. Theoretical Background

Since the expressivity of the language directly influences the scalability of the reasoning, OWL offers three sublanguages with different degrees of expressivity: OWL Full, OWL DL and OWL Lite. OWL Lite is the least expressive sublanguage and it provides the minimal subset of language features. It provides language constructs for hierarchy constructions for classes and properties, equality, property restrictions and characteristics (e.g., inverse, transitive properties), cardinality limited to 0 and 1 and intersection of classes. OWL DL (DL stands for “Description Logic”) extends OWL Lite with additional language constructs for expression of specific constraints for cardinality, disjoint, union and complement. Finally, OWL Full contains all OWL constructs and provides unconstrained use of all RDF constructs. Another key difference between the sublanguages is that OWL Lite and OWL DL are decidable\textsuperscript{10}, while OWL Full is undecidable.

2.1.1.3 SPARQL

SPARQL\textsuperscript{11} is a query language for RDF and it is recursive acronym for SPARQL Protocol and RDF Query Language. It is defined with a set of specifications which support querying and manipulation of RDF graph on the Web or RDF store. The two core specifications are: the SPARQL Query Language\textsuperscript{12}—defines the syntax and the semantics of the language, and the SPARQL Protocol\textsuperscript{13}—defines the method for conveying queries to a SPARQL service. In addition, there are also specifications on describing SPARQL services (SPARQL 1.1 Service Description)\textsuperscript{14}, updating RDF graphs (SPARQL 1.1 Update)\textsuperscript{15}, executing distributed queries (SPARQL 1.1 Federated Query)\textsuperscript{16}, representation of results in XML, JSON, TSV or CSV, and use of the HTTP methods for the purpose of manipulation with RDF graphs (SPARQL 1.1 Graph Store HTTP Protocol)\textsuperscript{17}.

A SPARQL query is defined as a set of triple patterns which are also known as graph patterns. A triple pattern is an RDF triple except that the subject, the predicate or the object can be a variable. A simple SPARQL query will contain a SELECT clause which identifies the variables to appear in the results set, and a WHERE clause which defines the graph pattern which should be matched against the queried RDF graph. SPARQL is an expressive language which supports optional query parts, filters, aggregation, nested queries and negation. The results from execution of SPARQL queries can be expressed as result sets using SELECT and ASK constructs, or as RDF graphs using the CONSTRUCT or DESCRIBE constructs. Listing 2.2 shows a SPARQL SELECT query to retrieve information about the mashups developed by Bob (cf. Figure 2.1).

\textsuperscript{10}Decidability - existence of a procedure which can provide complete reasoning.
\textsuperscript{11}https://www.w3.org/TR/sparql11-overview/
\textsuperscript{12}https://www.w3.org/TR/sparql11-query/
\textsuperscript{13}https://www.w3.org/TR/sparql11-protocol/
\textsuperscript{14}https://www.w3.org/TR/2013/REC-sparql11-service-description-20130321/
\textsuperscript{15}https://www.w3.org/TR/sparql11-update/
\textsuperscript{16}https://www.w3.org/TR/sparql11-federated-query/
\textsuperscript{17}https://www.w3.org/TR/sparql11-http-rdf-update/
2. Background and State-of-the-Art

```sparql
PREFIX ex: <http://example.org/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX dcterms: <http://purl.org/dc/terms/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT ?mashupURI, ?label
WHERE {
  ?mashupURI dcterms:creator ex:#Bob .
}
```

Listing 2.2: SPARQL query to get the all mashups developed by Bob.

The result from executing this query on the RDF graph from Figure 2.1 will return the following result set.

Table 2.1: Query results from the SPARQL query from Listing 2.2.

<table>
<thead>
<tr>
<th>mashupURI</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://example.org/#Visual-mashup">http://example.org/#Visual-mashup</a></td>
<td>Visual mashup</td>
</tr>
<tr>
<td><a href="http://example.org/#Photo-mashup">http://example.org/#Photo-mashup</a></td>
<td>Photo mashup</td>
</tr>
</tbody>
</table>

2.1.1.4 Linked Data

Existence of links among data is crucial for exploration of the Web of Data and retrieval of related data. By following those links a person or a computer can find other, related data. In July 2006, Tim Berners-Lee introduced “Linked Data” – an effort towards publishing and linking semantic data on the Web. In a nutshell, Linked Data defines recommended best practices for publishing, sharing and linking structured information on the Web. It encompasses principles and technologies for data sharing and re-use on a massive Web scale. Linked Data builds upon existing Web standards, such as the Uniform Resource Identifiers (URIs) and the Hypertext Transfer Protocol (HTTP). In 2006, Tim Berners-Lee in one of his personal architectural notes introduced a set of four principles, also known as the Linked Data principles:

1. Use URIs as names for things.
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF, SPARQL).
4. Include links to other URIs, so that they can discover more things.

---

18 https://www.w3.org/DesignIssues/LinkedData.html
The first principle endorses the use of URI references for unique identification of things on the internet such as Web documents, but also for identification of real objects and abstract concepts. The second principle advocates the use of the HTTP protocol as a unified access mechanism to the description of the resource identified with the URIs. By providing HTTP URIs for identification of the objects and concepts, a user can then dereference those URIs and retrieve description of the object or concept. The third principle advocates the use of RDF as a data model for describing and publishing data. The principle also encourages support of the SPARQL language so that information about an object can be queried. Finally, the fourth principle advocates provisioning of links to other things. This enables clients to navigate through the Web of Data and discover more things. For example, an RDF description of a movie could contain link to the person who directs the creation of the movie. By following this link, the client could retrieve description for the movie director. If the director is linked to other movies he directed, the client can further explore those movies as well.

Linked Data triggered also other initiatives with the aim on reducing the effort of publishing RDF and publishing semantic Web data under an open license. For example, the W3C RDB2RDF\(^20\) working group defined standard languages for mapping relational data into RDF and OWL.

Another community effort is the W3C Linked Open Data\(^21\) project which aims at

\(^20\)https://www.w3.org/2001/sw/rdb2rdf/
\(^21\)https://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData
publishing Linked Data data under an open license, which does not impede its reuse for free. The goal of the initiative is to stimulate collection of open data from various sources, provide semantization of the data using RDF, interlink the datasets and publish them as Linked Data. Following these guidelines triggered generation of huge amount of datasets which became part of the LOD cloud. Examples of open data sources that have become part of the LOD cloud include Wikipedia\(^{22}\), OpenStreetMap\(^{23}\), WordNet\(^{24}\) and many more data sources published under an open license. Figure 2.2\(^{25}\) shows an excerpt of the LOD cloud.

While in May 2007 the LOD cloud listed only 12 datasets, in the following years the number of published datasets has significantly increased\(^{26}\). From 45 datasets in 2008, to 95 in 2009, 295 in 2011, 1,091 in 2014, and 1,146 as of January 2017.

One of the reasons for such increase could be the additional set of principles introduced by Tim Berners-Lee in 2010\(^{27}\). In 2010, he extends the initial set of Linked Data principles and contributes five additional principles which relate to the Open Data initiative. These principles have been introduced under a so-called five-star schema, which is defined as follows:

- 1 star - Available on the web (whatever format) but with an open licence, to be Open Data.
- 2 stars - Available as machine-readable structured data (e.g. excel instead of image scan of a table).
- 3 stars - As 2-stars plus non-proprietary format (e.g. CSV instead of excel).
- 4 stars - All the above plus use open standards from W3C (RDF and SPARQL) to identify things, so that people can point at your stuff.
- 5 stars - All the above, plus link your data to other people’s data to provide context.

2.1.1.5 DBpedia

Providing links to other datasets is one of the key principles while publishing Linked Data. When publishing a Linked Data dataset, the dataset provider should link the contributed dataset with other relevant datasets from the LOD cloud. DBpedia\(^{28}\)\(^{37}\) is one of the most exploited datasets for linking. It is a multi-domain dataset which provides good basis for interlinking and integrating datasets with the LOD cloud. The DBpedia project’s goal is

\(^{22}\)https://www.wikipedia.org/
\(^{23}\)https://www.openstreetmap.org/
\(^{24}\)http://wordnet.princeton.edu/
\(^{25}\)Figure 2.2 is a modified version of an image retrieved from http://lod-cloud.net/. The image is licensed under a Creative Commons Attribution Attribution-ShareAlike 3.0 license.
\(^{26}\)This statistics are taken from the LOD Cloud document (http://lod-cloud.net/) which is updated on regular basis.
\(^{27}\)See “Is your Linked Open Data 5 Star?” at https://www.w3.org/DesignIssues/LinkedData.html
\(^{28}\)http://dbpedia.org/
2.1. Theoretical Background

to build large-scale, multilingual knowledge base by providing structural information extracted from Wikipedia. DBpedia primarily relies on information derived from Wikipedia articles. The information is extracted from the infoboxes, but also from tables, lists, and categorization data. The extraction process starts with a Wikipedia article which is parsed and using specialized extractors valuable information is extracted. Using pre-defined mapping templates, the information provided in infoboxes is mapped to a formally described DBpedia Ontology. The DBpedia instance data and the DBpedia Ontology are the key assets generated by the DBpedia project. The DBpedia release\( ^{29} \), announced on April 2016, consists of 754 classes, 1,103 object properties and 1,608 datatype properties. Moreover, the DBpedia Ontology is also aligned with other ontologies using 410 \texttt{owl:equivalentClass} and 221 \texttt{owl:equivalentProperty} mapping statements. As of April 2016, based on all supported Wikipedia editions, DBpedia describes over 6M entities and provides 9.5 billion information statements (RDF triples) for these entities.

2.1.2 Knowledge Extraction and Integration

Huge amount of digital data, in all shapes and sizes, is being generated at astonishing rates. In 2011, digital information has grown nine times in volume in just five years \cite{38} and by 2020, its amount in the world is estimated to reach 40,000 exabytes \cite{39}. According to a report from International Data Corporation (IDC) published in June 2014, unstructured content accounts 90\% of all the information \cite{40}. Unlocking the value hidden in the unstructured content is more critical than ever.

Text is one of the most common forms used for storing information. It includes emails, documents, social media content, Web pages, audio and video transcripts. To unlock the hidden value of the information stored in text based documents, information extraction techniques are being utilized. The goal of the Information Extraction (IE) \cite{41} is to automatically extract structured information from unstructured and semi-structured documents. The extracted information is only being useful if it is further formally represented as knowledge. In this section, we first introduce knowledge extraction, followed by knowledge integration techniques in the context of Semantic Web. Next, we describe the Named Entity Recognition (NER) technique, which is employed in the thesis and its role in knowledge integration. Finally, we briefly describe the Natural Language Processing Interchange Format (NIF) which provides means for representation of NLP results generated by an NER system.

2.1.2.1 Information and Knowledge Extraction

Named Entity Recognition. Named Entity Recognition (NER), also known as entity extraction or entity identification, is an IE task which aims at identification of entity mentions in texts \cite{42}. Formally, entities are defined as proper noun phrases which comprise of one or more tokens (proper nouns) in a text. Part of the NER task is also to classify

\( ^{29} \)Statistics for the 2016-04 DBpedia release: http://wiki.dbpedia.org/dbpedia-version-2016-04
the entity mentions into pre-defined set of categories (entity types) such as persons (PER), organizations (ORG), locations (LOC), geopolitical entities (GPE), etc. An entity type can help to disambiguate the entity mention and provide information whether the entity mention is referring to a company or a person which have the same name. However, there can be situations when there are identified two entity mentions with a same name and a same type, but they can refer to different real-world entities. For example, in a text can occur two entity mentions under the label “Maradona”, however, one can refer to the famous football player Diego Maradona, while the other to the football player Raúl Maradona. Thus, in such scenarios the entities can not be disambiguated based on the assigned entity types. In order to solve this issue, it is a current trend that disambiguation of entities is performed by assigning unique URI identifiers. The task of assigning URIs is also known as Entity Linking, which has been first formalized in [43]. These URI identifiers are usually re-used from knowledge bases which contain those entities. Wikipedia, DBpedia and Freebase are typical examples of such knowledge bases.

**Terminology Extraction.** Terminology extraction is an IE task with a goal to automatically extract terms from a given text [44]. These extracted terms are then used to build a terminological knowledge base, such as WordNet or BabelNet, or to describe the contents of the text document. The terminology extraction approaches usually rely on the output from NLP techniques, such as tokenization, Part-of-Speech (POS) tagging and noun phrase chunking, to identify term candidates. These term candidates are usually extracted as noun phrases, noun phrases preceded with adjectives or prepositional noun phrases.

**Relationship Extraction.** Very often there is need to identify how entities are related to each other. The relationship extraction [45] can provide information on the relationship(s) between two specific entities - how are they related or if they are in a particular relationship. Also, relationship extraction can identify all possible relationships between two types of entities, for example, between a “person” and an “organization”.

### 2.1.2.2 Knowledge Integration

Knowledge Integration (also known as Data Integration) has been classified as hard problem [46]. The data integration tasks becomes even bigger when the goal is to integrate large number of data sources. In a typical data integration scenario, there are different data publishers, which provide datasets using different data models and different schemas, however with an overlap in the content. A data consumer, would then need to align the schemas, detect duplicates and establish links so that the data is integrated. This integration scenario, should be replicated every time when a new customer wants to integrate and consume the data from the different datasets. This is a huge technical challenge for the customers, which can be solved by publishing the data as Linked Data according to the Linked Data principles. This includes, reuse of terms from well established vocabularies, publishing mappings between terms from different vocabularies and providing links between same and related entities.
2.1. Theoretical Background

Although knowledge integration can be technically solved via Linked Data, it still requires sophisticated approaches for discovering links within and among datasets, alignment of ontologies, and linking unstructured content (e.g. texts) with structured knowledge bases.

**Link Discovery.** Usually, the links within an RDF dataset and with other datasets are generated along with the creation of the dataset. However, very often some links within the dataset, or links with other datasets, might be missing. The Link Discovery task aims at discovery of typed links between given knowledge bases independent of the domain [47]. When performing interlinking, certain conditions must fulfill. These conditions can be based on similarity metrics and can consider the graph around the target instances, which can be addressed using a path-like selector language [48]. The discovered links can be of type `owl:sameAs`\(^{30}\) and relate individuals which have the same “identity”, but also RDF links of other types, such as `foaf:knows`\(^{31}\) to describe a relation between a person and another person that he or she knows.

**Ontology Alignment.** The Data Integration task requires not only integration at the instance level—discovery of links, but also integration at the schema level—establishing mappings between different vocabulary terms. The task of defining mappings among terms from different vocabularies is also known as Ontology Alignment or Ontology Mapping. The goal of the ontology alignment task is to establish relations between vocabularies (concepts and relations) defined by different ontologies [49].

**Linking Information from Unstructured Sources.** While the Link Discovery and Ontology Alignment techniques support the integration of structured information, there is also need to integrate unstructured content (e.g. texts) with structured knowledge bases. This can be achieved by linking the hidden information in the unstructured texts with a targeted knowledge base. In particular, information such as named entities and terms extracted in a text can be further linked with a given knowledge base. Entity Linking (EL) [43] is an NLP task of assigning URI identifier to an entity from a given knowledge base. Entity Linking is used to link entity mentions from an unstructured content (e.g., text or transcripts) to their representation in a structured knowledge base. Wikipedia, DBpedia, BabelNet and Freebase are one of the most exploited knowledge sources for entity linking.

Similarly, extracted terms from a text can be further linked with their representation in lexical knowledge bases, such as WordNet\(^{32}\) or Dbnary\(^{33}\).

Both approaches, dealing with linking entities and terms, must address the ambiguity. While entity mention can refer to more entities, also the correct sense of the term should be identified. In computational linguistics, the problem of determination of the sense (i.e. meaning) of the word/term in a sentence is also known as word-sense disambiguation.

\(^{30}\)https://www.w3.org/TR/owl-ref/#sameAs-def

\(^{31}\)http://xmlns.com/foaf/spec/#term_knows

\(^{32}\)https://wordnet.princeton.edu/

\(^{33}\)http://kaiko.getalp.org/about-dbnary/
2. Background and State-of-the-Art

2.1.2.3 Encoding results from Knowledge Extraction and Integration

In past years, several Linked Data compatible formats for encoding results from the Knowledge Extraction tools have been proposed. The NLP Interchange Format (NIF) [50] is an RDF/OWL-based format that aims to achieve interoperability between NLP tools, language resources and annotations. NIF is supported with an ontology which provides classes and properties for annotation of strings in texts and description of their relations. It also defines schema for minting URIs for strings. Using NIF, the results from the entity and terms extraction can be encoded. Further, using the Internationalization Tag Set (ITS 2.0)\(^{34}\) vocabulary, in particular with help of the `itsrdf:taIdentRef` property, a URI identifier from the target knowledge base can be assigned for the entity. Similarly, using the “lexicon model for ontologies” (Lemon)\(^{35}\) terms can be linked with their lexical entries in a terminological knowledge base.

2.1.3 Recommender Systems

In the past years, the popularity of the Recommender Systems (RS) has significantly increased. RS are extensively exploited in scenarios dealing with huge amount of information. In such scenarios, a RS is employed to filter out irrelevant and provide relevant recommendations for the end-user. There are two main approaches utilized for generation of recommendations [51]: collaborative filtering and content based filtering. In addition, there are knowledge based, demographic and hybrid approaches.

2.1.3.1 Collaborative Filtering

The Collaborative Filtering (CF) [52] approach is based on the user-item interactions. A CF based approach is generating item recommendations from users with similar tastes. A CF approach is developed in two phases. In the first phase, the similarity between the users is estimated. In the second phase, items from the top-N most similar users are recommended. In CF, the items are considered as black boxes. In other words, the content of the items is not considered, but only the user interactions with the items are used to generate recommendations. There are two types of collaborative filtering approaches, user-based and item-based. In an item-based CF, items similar to the items of interest are recommended to the user. In the user-based CF, items from similar users are recommended. The CF approaches are language-agnostic, and they can be applied for recommendation of content items in different languages, or items such as videos, images and music, which do not have content explicitly associated with them.

There are also several limitations with the CF approaches. Data sparsity is an important aspect for CF, since CF can only provide meaningful recommendations if sufficient amount of data is available. In other words, it requires reasonable number of users with overlapping interests. Thus, CF will fail to provide recommendations for new users or users.

---

\(^{34}\)https://www.w3.org/TR/its20/

\(^{35}\)https://www.w3.org/2016/05/ontolex/
with unique tastes. New items also suffer of not being recommended until a user interacted with these items. Due to a large data sparsity and the size of the user-item matrix, a CF system might also encounter scalability issues.

2.1.3.2 Content Based Filtering

A Content Based Filtering (CBF) [53] approach recommends items similar to the items of interest to the user. In CBF, each item is described with a set of features which are used to compare the similarity between the items. The features can be generated by analyzing the content associated with the items. Usually, text mining techniques are employed to extract features from text based content associated with the items. Features such as keywords, terms, phrases, entities or other type of useful information is extracted and further used as representation of the items. The extracted information is further normalized using various NLP techniques, such as stemmatization, lemmatization, stop word removal, etc. Finally, each item is represented using a feature vector, using a so called Vector Space Model (VSM) [54]. Each dimension of the vector represents a particular feature with an assigned feature weight. Using statistical analysis such as the TF-IDF [53] weighting scheme, weights of individual features are computed and assigned.

Content based filtering approaches have some advantages and limitations compared to the collaborative filtering based. The recommendations of the CBF approaches can be explained to the user by providing the features which caused the recommendations. On the contrary, CF systems act as black boxes and it is difficult to provide explanation for the generated recommendations. Further, CBF can effectively produce recommendations for new users as well as for new items. However, CBF requires sufficient amount of user-item interactions in order to provide relevant recommendations. Since, CBF approaches provide recommendations which match user preferences, they suffer from providing novel and serendipitous recommendations.

2.1.3.3 Hybrid and Other Approaches

In particular scenarios, recommendations based on the demographic information about the user are required [51]. For example, to recommend Web sites related to the country and the language of the user, or recommend restaurants near the user location. This type of RS are usually referred as demographic RS. Knowledge based RS are systems which generate recommendations with support of a domain specific background knowledge. For example, a user could express his preferences of building a mashup application with a particular functionality. By having a background knowledge consisting of API functional taxonomy, we can then easily estimate how well a particular API match the user needs.

A Hybrid RS usually combines two or more of the previously described approaches with the goal to eliminate particular deficiencies of the individual recommendation techniques.
2. Background and State-of-the-Art

2.1.3.4 Linked Data Recommenders

In the last decade, Linked Data has been identified as valuable source of information for building recommendation systems. In the context of Linked Data, recommendation techniques have been exploited in two ways. First, in direction of developing recommender systems “for Linked Data”. This type of approaches generate recommendations of type of Linked Data resources. The other type of Linked Data RS “use Linked Data” as background knowledge to boost the recommendations.

The main objective of a Linked Data recommender is to identify, rank and recommend Linked Data resources relevant for the user. In the traditional recommender systems, the Vector Space Model is the most exploited model for representation of user profiles and describing items. In the context of Linked Data RS, users and items are usually represented as nodes in a given RDF graph. Since Linked Data and RDF are in general based on graph based data model, many of the existing Linked Data recommendation approaches exploit concepts from the graph theory. Graph theory concept are used to represent the user profiles and the items as sub-graphs in a given RDF graph. The user profiles are represented as graph consisting of resources surrounding the user node and the relations among them. Similarly, the items can be also described as a subgraph with the resources connected to the item resource.

Also, graph theory based approaches are adopted to measure similarity between users and/or items in RDF graphs. In the last years, various measures and RDF resource similarity metrics have been proposed. These semantic measures are employed to “evaluate the degree of overlap between the entities based on a set of pre-defined factors, such as taxonomic relationships, particular characteristics of the entities, or statistical information derived from the underlying knowledge base” [55].

Section 2.2.3 provides an overview of the existing Linked Data recommender methods and systems.

2.1.3.5 Validation and Evaluation of Linked Data Recommenders

Evaluation Datasets. There are many datasets available for evaluation of recommender system, which can be also re-used for evaluation of Linked Data recommendation approaches. However, most of these datasets are not published as RDF (or Linked Data) and they do not provide URIs to Linked Data resources. Thus, very often there is need to map the information from these datasets, the items identifiers, to Linked Data URIs. In order to fill-in this gap, several datasets from the recommender systems research have been adopted for evaluation of Linked Data recommenders.

The MovieLens36 dataset has been generated based on a data collected from the MovieLens37 Web site. The latest version provides over 20M ratings for more than 20K movies and 138K items. The MovieLens is originally provided without links to Linked Data resources. In [56], the authors extended the dataset with links to DBpedia.

36https://grouplens.org/datasets/movielens/
37https://movielens.org/
2.1. Theoretical Background

MovieTweetings\textsuperscript{[57]}\textsuperscript{38} is a dataset which provides ratings for movies extracted from tweets on Twitter. The movie items from the dataset, have been also mapped to DBpedia URIs \textsuperscript{[58]}.

LibraryThing \textsuperscript{[56]} is another LOD dataset based on information from the LibraryThing Web site, an online catalog of books. The dataset provides ratings for books which have been mapped to DBpedia URIs.

Last.fm \textsuperscript{[56]} is a dataset from the music domain. It has been created based on an implicit knowledge about users and what they listen. The artists from the dataset have been mapped to DBpedia URIs.

In our work, we have developed specific dataset from the Web services domain, named Linked Web APIs dataset. This dataset, among the other possible use cases, can be also used for development of Linked Data recommenders. The dataset, contains information about developers, developed mashup compositions, and Web APIs used in these compositions. A unique feature of the dataset is the time information, which is an important piece of information for time-aware recommendation systems. Chapter 3 provides detailed description of the dataset.

Metrics. Linked Data recommenders are evaluated similarly as the traditional recommendation systems. In general, recommender systems are evaluated for their accuracy in generating recommendations and the usefulness of the recommendations.

Precision (P) and Recall (R) are the most popular metrics for evaluation of recommender systems. The precision is defined as the fraction of retrieved items that are relevant, while recall as fraction of the relevant items that are relevant. The precision and recall are usually used to evaluate the lists of recommendations with different sizes (N). For example, evaluation of recommendations @Top-N where N is 5, 10, 20, 50 or 100. The precision and recall are metrics of binary type. It is only evaluated if an expected item is present or not in the recommendation list. Thus, the relevance score and the ranking position of the items are not taken into account. In order to provide weighted average of the precision and recall, some works also use the $F1$ metrics for evaluation.

Mean Average Precision (MAP) metric is similar to F1 and evaluates the average precision across several different levels of recall or the average precision at the rank of each relevant item.

Normalized Discounted Cumulative Gain (nDCG), compared to precision and recall, measures the quality of the ranked list of items. It takes into account the relevance and the ranking of the items by assigning higher weight to items with higher rank. nDCG is also evaluated at different Top-N recommendation lists.

Mean Reciprocal Rank (MRR) is another metric which considers the rank position of the items in the list of recommendations. For each query, the MRR is computed as a reciprocal of the rank at which the relevant item is retrieved.

Although accuracy is an important aspect for a recommender system, it does not provide clear evidence whether the recommendations are effective and satisfactory. Thus, the

\textsuperscript{38}https://github.com/sidooms/MovieTweetings
usefulness of the recommendations in terms of “how surprising and diverse the recommendations are” is another important aspect.

Serendipity is the ability of a system to generate serendipitous recommendations. A serendipitous recommendation is an item which is assumed to be surprisingly interesting for the user. In an RDF graph, serendipity can be measured as the length of the shortest path from the user node to the recommended resource. A path with longer distance indicates a greater surprise. An overall serendipity of a set of recommended resources (set C) can be computed as the average serendipity of the resources in the set. While recommendations can be precise and surprising, it is also important to generate diverse recommendations.

Diversity is a quality of a recommender system to generate different recommendations. In other words, such system will generate diverse recommendations, for example, from different category, location or culture. In an RDF graph, the diversity of the recommended resources can be computed as the average dissimilarity among all resource pairs. In order to compute the (di-)similarity, we need to take into account the nodes in the surrounding graphs of the resources and consider any overlapping nodes. Both, the serendipity and diversity, can be evaluated at different Top-N recommendation lists.

2.2 Previous Results and Related Work

This sections, presents and surveys the work relevant to the topics and contributions of the dissertation thesis. The thesis contributes to three research fields. In particular, this section presents the related work in:

- **Semantic Web Services**: semantic Web service models and datasets in Section 2.2.1.
- **Named Entity Recognition and Entity Linking**: methods and systems for identification, classification and linking of Named Entities in Section 2.2.2. In Section 2.2.2.1 we also present related work on identification of salient entities.
- **Linked Data Recommenders**: methods and systems for efficient retrieval and utilization of Linked Data in Section 2.2.3.

2.2.1 Semantic Web Services

In the past decade, Web services have become widely accepted for flexible sharing and integration of system functionalities and data. Functionalities and data offered by one or more Web services, either manually or in an automated way, are combined in composite applications, also known as service mashups. Many works in this area have been conducted in order to enable automation of service provisioning tasks such as discovery, selection, composition and execution, through enhancing services with semantics, known as Semantic Web Services (SWS). The ultimate goal of the SWS is to enable automated provisioning of Web services on a semantic level. The main results in the SWS area includes
various semantic models and formal reasoning mechanisms that can support automated Web service discovery, selection, composition and execution.

Web service selection is a complex process where the best Web service that matches user preferences is selected from a set of service candidates provided by the service discovery process. Although the SWS research presented promising results, it has failed to gain a significant uptake on the Web and fulfill their goal on reducing manual effort required when using Web services. First and the main reason for their minimal impact is the high diversity of incompatible and complex solutions which requires additional effort from the users. And secondly, existing automation methods usually assume that the software agents can entirely replace users’ activities which leaves an open gap between the services and the users. They do not, or just partially take into account the services’ and the users’ social context in the service directories, such as the relationships among the users, services and service providers, the way they collaborate and communities they create. The social behavior of the users and services is a very essential kind of knowledge that can improve the results of the service provisioning.

This section surveys the literature on:

- Web Service Description Models: formal semantic and non-semantic models for description of Web services, and
- Web Service Data Sources: available datasets with Web service descriptions.

### 2.2.1.1 Web Service Description Models

In the last decade, Web services have been under an active development with focus on formal models for describing Web services. In general, two types of Web service description models have been investigated: **syntactical models**-describe how to consume a Web service and exchange data in order to provide syntactic interoperability, and **semantical models**-provide semantic meaning or semantic constraints on the data, as well as semantic meaning of particular aspects such as the functional, non-functional, provenance, temporal and technical aspects. Below we briefly introduce the major Web Service description models; the semantic, followed by the syntactical models.

**OWL-S**\(^{39}\), also known as **Semantic Markup for Web Services** [5], was the first major semantic Web service model with the goal to enable users and software agents to discover, invoke, compose, and monitor Web services. OWL-S defines three ontologies: **service profile**-dedicated for service advertisement and discovery, **process model**-for description of the Web service operations and behavioral aspects, and **service grounding**-for description of the physical access to the Web services. OWL-S can be considered as top-down conceptual model for semantic Web services. In summary, OWL-S captures the functional and non-functional service properties via the service profile ontology, behavioral properties using the process model ontology and technical properties via the service grounding ontology. Unfortunately, OWL-S does not capture neither provenance (who created or used the

\(^{39}\)https://www.w3.org/Submission/OWL-S/
service) nor temporal information (i.e. the time when it was created, updated, etc.). Most importantly, OWL-S is very complex for use from the perspective of an average service developer [10].

**Web Service Modeling Ontology (WSMO)** [6] is an ontology which aims at describing all relevant aspects related to general services which are accessible through a Web service interface. WSMO provides means to describe Web services that provide access (searching, buying, etc.) to specialized services. It provides ontological specifications for the core elements of semantic Web services. WSMO, same as OWL-S, is a top-down model for describing Web services. In summary, via four ontologies WSMO captures the data model, functional, non-functional and behavioral service aspects. Although highly expressive, WSMO does not capture the temporal aspects. One of the main issue with the WSMO service model is that it requires significant manual effort to develop the models which also requires well trained experts [11].

**Semantic Annotations for WSDL and XML Schema (SAWSDL)** [40] defines extension attributes for annotation of WSDL documents. These attributes allow to link individual elements of a WSDL document with their semantic description. Thus, SAWSDL provides bottom-up approach for modeling Web service descriptions. It defines three attributes for semantic annotation: modelReference to point to semantic concepts that describe a WSDL element, and loweringSchemaMapping and liftingSchemaMapping to specify mappings between an XML data and the semantic information model. In summary, SAWSDL does not specify a language for representing the models, but a mechanism for referencing semantic concepts and WSDL and XML schema elements. One of the key problem with SAWSDL is that it is exclusively targeting WSDL/SOAP based Web services which are nowadays not prevalent on the Web [12].

**WSMO-Lite** [41], also known as *Lightweight Semantic Descriptions for Services on the Web* [7], provides annotation mechanism based on RDFS, which can be used to annotate WSDL documents. While SAWSDL can be used to link WSDL elements with semantic concepts, using WSMO-lite we can describe the functional, non-functional and behavioral service components. WSMO-Lite, same as the SAWSDL, can be classified as bottom-up Web service modeling approach. WSMO-Lite captures the functional, non-functional, and informational model semantics of the Web services. However, it does not capture the provenance and the temporal information. Moreover, WSMO-Lite has been developed for WSDL services, which are nowadays not prevalent on the Web [12].

Nowadays, vast amount of Web Services and Web APIs are described with text in HTML documents. Such descriptions are, however, not machine-readable and they can only be processed by humans. **hRESTS** [8] addresses this issue and defines a poshformat [42] for machine-readable descriptions of Web APIs. Using the microformat [43] mechanism, hRESTS
can be used to embed annotation in already existing HTML pages describing Web APIs. It can be used to describe a service, its operations, inputs and outputs. In summary, hRESTS captures the functional, non-functional, technical aspects and the information model. Temporal and provenance information are, however, not considered by hRESTS.

Another problem is that microformats, such as hRESTS, face also with scalability issues due to the lack of namespace support.

Semantic Automated Discovery and Integration (SADI) \[9\] defines design patterns for an automated discovery, composition and orchestration of Web services. SADI requires that the service interfaces are described using OWL-DL classes, in order to define required inputs and outputs for the service. The development of SADI has been driven by requirements from the bioinformatics domain. SADI can be used to capture functional and technical aspects, but there is lack of support for the non-functional, behavioral, provenance and temporal information.

Hydra\[44\] defines a vocabulary for hypermedia-driven Web APIs. The main goal of the Hydra vocabulary is to provide valid state transitions for clients. Clients can use this information to construct queries on-the-fly instead of hardcoding them into the client at the design time. Hydra is an RDF vocabulary which defines classes and properties for description of Web APIs. Using the Hydra vocabulary an API provider can describe a Web API by providing its title, description, main entry point, the list of resources exposed by the API and the available operations. The Hydra specification is being developed by the Hydra W3C Community Group, however, it is not a W3C Standard nor on the W3C Standards Track. In summary, Hydra defines vocabulary for creation of hypermedia driven Web APIs. It primarily defines classes and properties for description of the service behavior (i.e. sequencing of operation invocations), functional, non-functional and technical properties. However, provenance and temporal information is out of the scope.

In the following text, we discuss the major syntactic Web service models, WSDL, OpenAPI and Apis.json.

WSDL (Web Services Description Language)\[45\] is a well known Web service description language. The first 1.0 version of WSDL has been published in 2000, while the WSDL version 2.0 was also endorsed by W3C in 2007 as a W3C recommendation. WSDL is an XML language for describing Web Services. It defines a language which can be used to describe the functionalities offered by a Web Service. WSDL is mostly used in combination with the SOAP protocol\[46\] and the XML Schema\[47\]. WSDL, in combination with SOAP, is primarily used for integration of enterprise systems, while on the Web, most of the Web applications expose their functionalities and data through Web APIs; primarily based on the HTTP protocol and the REST architecture.

OpenAPI\[48\] is a framework which supports the development of Web APIs throughout

\[44\]http://www.hydra-cg.com/spec/latest/core/
\[45\]https://www.w3.org/TR/wsdl120/
\[46\]https://www.w3.org/TR/soap12/
\[47\]https://www.w3.org/TR/xmlschema-0/
\[48\]https://www.openapis.org/
their entire development lifecycle. The OpenAPI specification is based on the `Swagger`\(^{49}\) specification. The development of Swagger started in 2010, while in 2015 it was rebranded as OpenAPI. The OpenAPI specification defines mechanism for description and documentation of RESTful APIs. An API description conforming to the OpenAPI specification is a JSON object, which may be serialized as JSON or YAML. Based on the OpenAPI description document can be generated a client code, documentation and/or test cases. The development of the OpenAPI specification is driven by the Open API Initiative, which is recognized as an open governance structure under the Linux Foundation. Swagger is not related to RDF and its syntax does not involve any type of semantics, as understood by the Semantic Web community.

`APIs.json`\(^{50}\) is a machine readable format for description of Web APIs. APIs.json format has been designed for public deployment and consumption by automated software agents. A software agent, with help of an APIs.json description document, can then discover and consume an API. Providers can use APIs.json to describe their APIs and its operations and publish the APIs.json description document in the root of their API endpoint. According to the APIs.json specification, the API description document is a JSON document which provides basic information for the API - its name, description, image and associated tags categorizing the API, time information - when it was created or last modified, links to the base endpoint URL and version number of the API the description refers to. A single APIs.json document can provide description for a collection of several APIs.

The major issue with the WSDL, OpenAPI and APIs.json Web service models is that they lack semantics, thus the Web API descriptions are not contextualized, they are not linked, it is difficult to integrate and query them.

In our work (see Section 3.2), none of the above-mentioned solutions can be used to completely capture the Web API information we have collected (see Section 3.2), they are complex, or they address service architectural models which are nowadays not prevalent on the Web. Thus, we have decided to develop a light-weight ontology on top of existing and well-established ontologies and appropriately extend them (see Section 3.3).

### 2.2.1.2 Data Sources

In order to increase the visibility of the Web APIs and attract more consumers, the Web API providers register and advertise their APIs in repositories. There are several repositories, which provide Web API descriptions. We bellow list the major data sources with Web API descriptions.

**ProgrammableWeb.com**\(^{51}\) is the largest Web API and mashup directory. It adopts characteristics of a social Web platform where Web API providers can publish and share information about their offered Web APIs and consequently increase their visibility. The API directory also allows developers to find appropriate APIs for their projects, or they can learn from showcases of existing mashup applications. As of July 2017, ProgrammableWeb

\(^{49}\)http://swagger.io/specification/

\(^{50}\)http://apisjson.org/

\(^{51}\)https://www.programmableweb.com/
2.2. Previous Results and Related Work

Table 2.2: Summary of Web Service Models and Data Sources. Data collected as of July 2017.

<table>
<thead>
<tr>
<th></th>
<th>PW</th>
<th>APIs.io</th>
<th>APIs.guru</th>
<th>API For That</th>
<th>Exicon</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Num of APIs</strong></td>
<td>17,768</td>
<td>1,103</td>
<td>480</td>
<td>612</td>
<td>1,900</td>
</tr>
<tr>
<td><strong>Format</strong></td>
<td>HTML</td>
<td>API.json</td>
<td>OpenAPI</td>
<td>HTML</td>
<td>HTML</td>
</tr>
<tr>
<td><strong>Access mechanism</strong></td>
<td>n/a</td>
<td>custom API</td>
<td>custom API</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td><strong>Access options</strong></td>
<td>n/a</td>
<td>list, simple search, add</td>
<td>list</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td><strong>Semantics</strong></td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
</tr>
</tbody>
</table>

lists over 17,000 APIs. For each Web API, it provides its title, short summary describing its functionalities, assigned tags and categories, non-functional properties such as its homepage, usage limits and usage fees, and technical information, such as supported formats and protocols. ProgrammableWeb also provides information about mashups and developers/consumers of Web APIs and mashups. The Web API and mashup descriptions are provided in HTML.

**APIs.io**[^53] is a Web API repository and search service. It uses the APIs.json description format (see Section 2.2.1.1). Web API providers can add API to APIs.io by providing URL to an API description document in JSON, which conforms to the APIs.json format. As of July 2017, APIs.io lists over 1,000 APIs submitted by over 200 users.

**APIs.guru directory**[^54] is a repository of Web API descriptions. It allows API providers to register their API to the repository and make it available through the directory. The APIs.guru directory also provides an API for listing the directory and retrieval of their description in the Swagger (OpenAPI) format. As of July 2017, APIs.guru lists almost 500 APIs.

**API For That**[^55] is another Web API directory which provides descriptions for over 600 APIs (as of July, 2017). A Web API is described with a basic information such as its title, short description, associated tags, category and link to the API homepage. API For That provides the API descriptions in the HTML format.

**Exicon**[^56] is an API directory which lists over 1,900 APIs. For each Web API it provides basic metadata information such as its title, short description, the category it belongs to, homepage, pricing information, supported protocol and data formats. The Exicon directory provides the API descriptions in the HTML format.

Although Web APIs enjoy significant increase in popularity, there are several problems related to the API descriptions which are starting to arise. Table 2.2 summarizes the information on Web service data sources. It can be observed that most of the Web APIs

[^52]: A mashup - a composition of one or more Web APIs.
[^53]: http://apis.io/
[^54]: https://apis.guru/openapi-directory/
[^55]: http://www.apiforthat.com/about
[^56]: https://app.exiconglobal.com/api-dir/
are described in HTML, in a non-machine processable format and there is limited access to the API descriptions.

2.2.1.3 Research Gaps

According to the review presented in Section 2.2.1.1 and Section 2.2.1.2, we identified following research gaps (RG):

- **RG1.1: Ad-hoc API description solutions without semantics.** Different Web API providers/aggregators use different ad-hoc properties to describe APIs and publish the API descriptions. In other words, there is no standardized procedure on what properties to use when describing Web APIs. Although there are efforts and initiatives such as OpenAPIs, yet most of the APIs are described in an ad-hoc manner. Moreover, there is no significant repository which provides Web API descriptions enhanced with semantics.

- **RG1.2: Lack of descriptions in a machine-readable format.** Vast amount of Web API descriptions are nowadays provided in a non machine-readable format, usually in HTML.

- **RG1.3: Web API descriptions provide limited contextual information.** Many of the API descriptions are missing contextual information, such as information on applications that consume the API, developers that developed the API, the time when the API was developed/modified/used, links to external knowledge bases (e.g., Wikipedia), etc.

- **RG1.4: Complex semantic web service models with partial Web service modelling capabilities.** Most of the Web service models consider only few of the Web service aspects, such as the functional, non-functional, behavioral, technical, temporal and provenance aspects. Moreover, some of existing service models are too complex for the service developers or target Web service models which are nowadays not prevalent on the Web.

All these problems make the discovery, sharing, integration, and assessment of their quality and consumption problematic. The lack of shareability and reuse are also reflected in higher maintenance and development costs of the Web APIs.

**Progress beyond the state-of-the-art.** In order to address the aforementioned research gaps, we have collected comprehensive information about Web APIs and created the largest dataset with semantic Web API descriptions. We have developed a light-weight ontology for modeling relevant Web API information, which builds on top of several existing and well established ontologies and appropriately extends and aligns them (RG1.1, RG1.4). The ontology addresses the gaps of the existing ontologies and captures all Web service aspects including functional, non-functional, technical, temporal and provenance information. The ontology has been used to semantically enhance the Web API descriptions (RG1.1) and provide them in a machine-readable format (RG1.2). The dataset is published as Linked
Data and it can be programmatically accessed via a dedicated SPARQL endpoint. Currently, it is the largest and most comprehensive dataset with Web API descriptions which are contextualized, can be referenced, re-used and combined (RG1.3). The ultimate goal is to establish the Linked Web APIs dataset as a central Linked Data hub for Web API descriptions. Chapter 3 describes the dataset, the process of creation, the supporting ontology and discusses the quality and the usefulness of the dataset.

2.2.2 Named Entity Recognition and Entity Linking

One of the first research papers in the field of Named Entity Recognition was presented by Rau in 2001 [42]. The paper presents an approach for automatic extraction of company mentions based on heuristics and handcrafted rules. While such approach could achieve relatively good precision, due to the handcrafted patterns and heuristics the approach is not a scalable solution. On the other hand, several approaches based on Supervised Learning for NER have been proposed. In general, these approaches leverage manually labeled corpora, which is then used to train a model. These NER supervised approaches have been based on Hidden Markov Models (HMM) [59], Decision Trees [60], Maximum Entropy Models [61], Support Vector Machines [62] and Conditional Random Fields [63].

One of the key problems with the NER approaches based on Supervised Learning is that they are dependent on the availability of the training data. The training data should be comprehensive and provide enough information for learning, validated by humans, up-to-date in order to train temporary valid models, multilingual in order to train NER models for different languages, structured so it can be processed by third party agents, and cover information from multiple domains.

Recently, the availability of large knowledge resources, such as Wikipedia, and knowledge bases developed on top of them, such as DBpedia [64], YAGO [65] and BabelNet [66], have triggered the emergence of the Entity Linking (EL) task. The ultimate goal of EL is to link entity mentions in a text with a given knowledge base. Linking words to Wikipedia, also known as Wikification, has been been first tackled by Mihalcea and Csomai in 2007 [67]. The EL methods primarily differ in the approaches used for candidate selection and candidate ranking. Name dictionaries based techniques are one of the most exploited techniques for candidate generation which are exploited by many EL methods [68, 69, 70]. Wikipedia provides useful information for generation of entity candidates. This includes Wikipedia pages—article titles are considered as the most common entity name, redirect pages—to collect different name variations (synonyms, abbreviations) for a same name, disambiguation pages—to extract abbreviations and other entity aliases, bold phases—to collect abbreviations and aliases, and Wikipedia hyperlinks—the anchor text of the link considered as name variation for the entity. To retrieve a list of candidates from a constructed dictionary a full label match in the dictionary can be executed for the entity mention. There are also methods which apply partial match when generating candidates [71, 72]. Since some entity mentions can occur as acronyms, some systems apply surface form expansion techniques to generate other naming variations. These techniques vary from heuristics based [73, 68] to a supervised learning based methods [74]. Finally, there are works which
propose utilization of search engines, such as Google [71, 72] and Wikipedia [69], to build a candidate lists.

Candidate ranking is the final stage in EL which ranks the entity candidates from the knowledge base according to their similarity to the entity mention. Existing techniques leverage features which can be context-dependent and context-independent. As for the context-independent features, there are methods which rely on the “entity candidate name”–“entity mention” string similarity [72], popularity of the entity in the knowledge base defined as graph metric (in/out degree, Google’s PageRank) [72], Wikipedia view statistics [75], or the type match of the entity mention and the candidate entity [72, 71].

There are works which also consider the context when ranking the entity candidates where the context is represented as a bag of words [75, 76] or a vector consisting of concepts [69, 70, 71, 72]. In addition to the textual information around the entity mention, as contextual information are also considered the other entity mentions that occur in the same document [75, 70]. The assumption is that entities co-occur in similar contexts.

For candidate ranking are utilized supervised and unsupervised methods. With regards to the supervised methods, existing solutions formulate the ranking problem as binary classification problem [73, 69, 71], learning to rank problem [68, 74, 72] and graph analysis problem [16, 77, 78]. As for the unsupervised ranking methods, in [70] the authors employ the unsupervised Vectors Space Model (VSM) while in [73] the authors solve EL by employing Information Retrieval methods where each entity candidate is individually indexed. Next, for each entity mention a query with its contextual document is executed.

Over the last few years NER and EL have triggered a development of several NER and EL systems. Bellow we list the most significant achievements.

- **DBpedia Spotlight** [13] is a widely used entity annotation system. It automatically annotates entity mentions in texts and links them to DBpedia resources. DBpedia Spotlight has been exclusively developed to perform linking only with the DBpedia knowledge base. It requires training data for execution of the entity recognition, classification and linking. It makes use of a surface form index generated from Wikipedia article text which is then exploited for entity spotting and linking. Entities are spotted and linked only if they are present in the DBpedia knowledge base. In other words, entities which are not present in DBpedia, are not recognized at all. Entity classification is not implemented as part of the DBpedia Spotlight system, but the types are retrieves from the DBpedia knowledge base, collected via Wikipedia infobox mappings. DBpedia Spotlight returns a list of spotted and linked entities in the document with associated confidence scores. However, the salience of the entities with respect to the document is not accessed; the entities are not analyzed in order to identify those entities that play an important role in a story that the document describes.

- **AIDA**[^AIDA] [14, 78] is another entity spotting and disambiguation system. Spotted enti-

ties are linked to their representation in the YAGO2 [65] knowledge base. Entities are spotted using the StanfordNER [63] system which, however, requires training data for development of the model for NER. AIDA is able to spot entities in English texts. AIDA, same as DBpedia Spotlight, does not assess the salience of the entities in the document.

- **NERD**[^58] [15] is a Web-based system supported with a framework which enables aggregation of several named entity extractors. The alignment among different entity types is realized using the NERD ontology[^59], which aligns the taxonomies of these tools. The salience of the recognized entities is not assessed.

- **NERD-ML** [79] is an entity recognition approach initially designed for extracting entities from tweets. The approach is based on machine learning classification of entity types based on a set of linguistic features. The set of features are retrieved from a trained CRF classifier and the output from NER systems which are already integrated in the NERD framework. This approach requires training data and also do not assess the importance of the recognized entities in the document.

- **Babelfy**[^60] [16] is a multilingual Word Sense Disambiguation and Entity Linking annotation system. It uses the BabelNet [66] semantic network as background knowledge base. It links individual fragments from the text to the BabelNet knowledge base. Babelfy performs only entity spotting and entity linking, while entity classification is not supported. Same as DBpedia Spotlight, it requires training data and only entities that are found in the BabelNet knowledge base are recognized and linked. Babelfy returns three types of scores, including relevance score, but no entity salience information.

- **FOX** [17] is a federated knowledge extraction framework for extraction of RDF from text. It is based on Ensemble Learning and exploits several NLP algorithms to extract entities. FOX integrates AGDISTIS [77] for disambiguation of entities. AGDISTIS is a graph-based approach which employs the HITS algorithm to find authoritative candidates for the discovered named entities. Both, FOX and AGDISTIS require training data in order perform entity recognition and linking. In comparison to DBpedia Spotlight, AIDA and Babelfy, FOX can recognize entities which are not found in the target knowledge base. Identification of entity salience is out of the scope in FOX.

- **StanfordNER**[^61] [63] is an NER system implemented in Java. It labels sequences of words in a text that represent a mention of an entity. StanfordNER is based on the Conditional Random Field (CRF) statistical modeling method. Currently, it is

[^58]: http://nerd.eurecom.fr/
[^59]: http://nerd.eurecom.fr/ontology
[^60]: http://babelfy.org/
[^61]: https://nlp.stanford.edu/software/CRF-NER.shtml
distributed with CRF sequence models for English, German, Spanish and Chinese. Entity salience is not evaluated.

- *AlchemyAPI*[^62] is a commercial system for entity extraction and classification. AlchemyAPI defines a taxonomy for classification of entity types[^63]. It also supports 8 languages including English, German, French, Italian, Portuguese, Russian, Spanish and Swedish. The methods used for NER/EL are undisclosed. AlchemyAPI returns relevance score for each recognized entity, but no salience information. In our work, we are interested in evaluation of entity salience, i.e. how important is the entity in the story that the document describes, which is different than the notion of “entity relevance”.

- *OpenCalais*[^64], currently owned by Thomson Reuters[^65], is an online tagging engine for entities, relationships, facts, events and topics. It supports disambiguation and linking across all processed documents, however there is no implementation information provided. OpenCalais supports English, French and Spanish. It provides relevance score for each recognized entity, but no salience information.

### 2.2.2.1 Identification of Salient Entities

In long documents, the list of recognized entities can be very long. Such list could contain entities which play an important role in a story that the document describes, but also entities which are not central for the document. Evaluation of entity salience is a relatively new research problem and hereby we present the related work on identification of salient entities.

In [18], the authors propose a supervised model for learning document aboutness through identification of salient entities. In the work, the authors use behavioral signals to collect salience annotations. Also, the features used for learning salience are computed from properties of the entity and the document corpus. In this work, features derived from graph analysis of an external entity knowledge graph are not considered. Also, the data used for training has not been manually checked.

In another work [80], the authors represent the aboutness of the documents using words and phrases that best reflect the central topics of a document. In other words, the aim of this work is not on identifying a list of salient entities, but on identification of words and phrases that reflects the central topics in the document. The training data used in this method is generated from an implicit user feedback derived from the search engine click logs.

Representing aboutness of microposts (tweets) by linking the tweets with semantically related Wikipedia pages has been proposed in [81]. The method described in [81] aims at describing the aboutness of a tweet as a set of concepts which are “contained in, meant

[^62]: http://www.alchemyapi.com/
[^63]: http://www.alchemyapi.com/api/entity/types.html
[^64]: http://www.opencalais.com/
[^65]: https://www.thomsonreuters.com/
by, or relevant” to the tweet. Thus, a tweet can be described also with entities which are relevant, but do not play an important role in the document. In our work, we aim only at identification of entities which play an important role in the story the document describes.

In [1], the authors propose a method which is exclusively designed for identification of salient entities. The model based on features derived from a coreference resolution system and additional background information. The model considers local features (e.g. the sentence index of the entity mention), and background knowledge by analyzing the centrality of each entity in the document (i.e. PageRank score). The entity centrality is evaluated on an entity graph containing all entities which are present in the document. As features which capture the background knowledge, is only considered the PageRank graph measure; other graph measure, such as HITS, in-degree, out-degree, etc., are not incorporated.

Development and evaluation of a method for identification of entity salience requires a dataset with salience annotations. There are two datasets with entity salience annotations that have been published recently, the “Microsoft Document Aboutness” (MDA) [18] and the “New York Times” (NYT) [1] dataset. The MDA dataset consists of entities occurring in randomly sampled Web pages (from head and tail distributions) together with a salience assessment for the entities. The NYT dataset is another dataset which has been created as an extension of the New York Times dataset. Although valuable, there are several problems with these datasets. Both datasets, the MDA and NYT, do not provide the documents due to copyright restrictions. Also, in both datasets, the entity mentions have been generated automatically using a proprietary NER system and they have not been checked by human. As a consequence, the datasets contain incorrect entity annotations. Furthermore, the salience annotations in the NYT dataset have been automatically generated by aligning the entities in the abstract and the document under the assumption that every entity which occurs in the abstract is salient, which is questionable. Overall, the design of these datasets does not foster their straightforward use and efficient development of methods for identification of salient entities.

2.2.2 Research Gaps

NER and EL enjoy significant increase in popularity and usage in the last decade. They have become the core technology for transforming unstructured data into structured. Table 2.3 provides summary of the most prominent NER and EL approaches. According to the review of the major solutions, we have identified following research gaps (RG):

- **RG2.1: Entity salience is not assessed by the existing NER/EL systems.**
  In long texts, the number of recognized entities can be very large. While some of the entities can be central for the document, the document can contain also entities which are not central in the story the document describes. Currently, there is no NER system which provides information on the salience/importance of the recognized entities. OpenCalais, AlchemyAPI and Babelfy returns relevance scores, but
in our work we exclusively focus on the salience of the entities, rather than on their relevance. In other words, an entity can be relevant but not central for the document.

- **RG2.2: Lack of a complete, publicly available and evaluated by human entity salience dataset.** Currently, there is lack of a complete, publicly available and evaluated by human dataset for learning entity salience. Existing datasets (NYT and MDA) face with several problems: they do not provide the documents due to copyright restrictions, the entity annotations have been generated automatically using a proprietary NER system and as a consequence contain incorrect annotations, and more importantly, the salience annotations have been generated automatically and have not been manually checked.

- **RG2.3: Supervised NER and training data.** Currently, vast majority of the NER systems are highly dependent on training data. DBpedia Spotlight and FOX require training data for entity recognition and linking, Bebelfy for linking, while StanfordNER requires data for its entity recognition and classification models. Acquisition of high quality training data is, however, usually expensive and relatively difficult task.

*Progress beyond the state-of-the-art.* In order to address the aforementioned research gaps, we have developed methods for unsupervised entity spotting and classification (RG2.3) for different languages. The methods have been implemented as part of an open-source NER system, named *Entityclassifier.eu*. We also developed a method for identification of salient entities (RG2.1, see Section 4.2). The method is supported with a novel dataset with entity salience annotations (RG2.2). Our work on knowledge extraction and integration with salient named entities, which addresses the aforementioned research problems, is described in Chapter 3.
Table 2.3: Named Entity Recognition and Entity Linking Systems.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Supported NER/EL tasks</strong></td>
<td>spotting, linking</td>
<td>spotting, classification, linking</td>
<td>spotting, classification, linking</td>
<td>spotting, classification</td>
<td>spotting, linking</td>
<td>spotting</td>
<td>spotting, linking</td>
<td>spotting, linking</td>
</tr>
<tr>
<td><strong>Method</strong></td>
<td>Context-based, VSM and TF-IDF</td>
<td>Graph-based</td>
<td>Undisclosed</td>
<td>Undisclosed</td>
<td>Conditional Random Fields</td>
<td>Graph-based</td>
<td>Machine learning</td>
<td>Graph-based</td>
</tr>
<tr>
<td><strong>Supported languages</strong></td>
<td>multilingual</td>
<td>n/a</td>
<td>EN, FR, ES</td>
<td>EN, DE, FR, IT, PT, RU, ES, SE</td>
<td>EN, DE, ZH, ES</td>
<td>multilingual</td>
<td>EN</td>
<td>multilingual</td>
</tr>
<tr>
<td><strong>Types</strong></td>
<td>fine-grained (derived from DBpedia)</td>
<td>fine-grained</td>
<td>fine-grained</td>
<td>fine-grained</td>
<td>coarse-grained</td>
<td>n/a</td>
<td>coarse-grained</td>
<td>n/a</td>
</tr>
<tr>
<td><strong>Granularity</strong></td>
<td>fine-grained</td>
<td>fine-grained</td>
<td>fine-grained</td>
<td>fine-grained</td>
<td>coarse-grained</td>
<td>n/a</td>
<td>coarse-grained</td>
<td>n/a</td>
</tr>
<tr>
<td><strong>Supported KBs</strong></td>
<td>DBpedia</td>
<td>YAGO</td>
<td>Thomson Reuters</td>
<td>DBpedia, YAGO, Freebase, GeoNames and more.</td>
<td>n/a</td>
<td>BabelNet, DBpedia</td>
<td>n/a</td>
<td>DBpedia</td>
</tr>
<tr>
<td><strong>Relevance and Salience</strong></td>
<td>n/a</td>
<td>n/a</td>
<td>relevance score</td>
<td>relevance score</td>
<td>n/a</td>
<td>relevance score</td>
<td>n/a</td>
<td>n/a</td>
</tr>
<tr>
<td><strong>Availability</strong></td>
<td>open-source under Apache License 2.0</td>
<td>open-source under CC BY-NC-SA 3.0</td>
<td>commercial</td>
<td>commercial</td>
<td>open-source under GPLv2 license</td>
<td>commercial under GPLv3 license</td>
<td>open-source under GPLv3 license</td>
<td>open-source under GPLv3 license</td>
</tr>
</tbody>
</table>
2.2.3 Linked Data Recommenders

In recent years, the use of Linked Data in the context of recommender systems has received increased attention. One of the first Linked Data recommendation approaches have been proposed in [30, 21]. In [30] the authors present method named dbrec, which defines a Linked Data Semantic Distance measure for measuring relatedness between resources. The measure takes into account both, the direct and indirect in/out links. The method relies only on instance data and links among considered instances. The method has been validated on a subset from DBpedia, which covers the music domain. It has been also applied on other domains, however, it requires manual pre-processing of the dataset. In [21] the authors describe the use of collaborative filtering based approach which exploits binary connections between users and items. The user-item matrix is constructed based on data retrieved via SPARQL CONSTRUCT queries which should be defined in advance for the particular domain and dataset. Thus, it requires expertise and background knowledge for the particular dataset and domain. The method has been applied on data from the music domain acquired from the Smart Radio service, DBTune MySpace and DBpedia. While the methods described in [30] and [21] have been applied on a single domain, [24] propose a Linked Data approach for cross-domain item recommendations. The recommendation method is based on a graph based weight spreading mechanism. The method has been validated on the music and geography domain. The method, however, requires human expertise for the given dataset and domain to select the relevant portion of data and to assign importance values for concepts and relations. In [25] the authors describe a graph-based recommendation approach. It relies on memory- and model-based link prediction techniques. It was validated on the music domain and it uses subset from the Freebase dataset as source for enrichment of user profiles. The method requires human expertise in order to assign weights on the edges in the network. A hybrid approach based on the collaborative filtering and content based filtering is proposed in [27]. The method also integrates a user diversity model in order to detect user propensity towards specific topics. Method was developed for the EventMedia [82] dataset from the music domain. The dataset was further enriched with information derived from DBpedia. A subset from DBpedia (categories) is exploited in order to enrich the topics of an event using the DBpedia topics of the involved artists. A semantic model named SemanticsSVD++ is proposed in [28]. The method captures the semantic taste evolution of users over time. The model is based on Matrix Factorization and it has been validated on the MovieLens and MovieTweetings datasets. A subset from DBpedia, movie categories, has been been used to enrich the datasets. In particular, the movie categories retrieved from DBpedia are integrated into the model. The method has been specialized for the movies domain and datasets from this domain. In [23], the authors propose a Linked Data enabled content-based movie recommender. It uses a vector space model to compute similarities between movies. The method has been developed for the movies domain [22] and its adaptation to other domains requires manual pre-processing of the dataset. It relies on knowledge retrieved from the LOD cloud such as categories from DBpedia and genres from Freebase and LinkedMDB. Thus, application of the method on other dataset/domain requires human intervention.
content based and collaborative filtering approach implemented as extension of SPARQL has been proposed in [26]. The recommendation engine extends the syntax and semantics of the SPARQL language. It enables development of customized recommenders over an arbitrary RDF graph. However, the method requires to specify the appropriate features. When using the collaborative filtering approach it is required to specify the path between the users and the items. For a content based approach, it is required to specify the features which describe the items.

A very related research topic to Linked Data Recommenders is the topic of Exploratory Search Systems (ESS) [31]. The Lookup Explore Discovery (LED) is among the first exploratory search systems. It recommends DBpedia resources related to the named entities recognized in the query. The resource ranking is supported by the DBpediaRanking algorithm [83]. The resources are ranked according to their co-occurrence with the resources’ labels in DBpedia abstracts, wikilinks information, as well as external information sources are queried (Google, Yahoo! and Bing) and their co-occurrence in the returned result pages is also evaluated. From the LOD cloud, only a small portion of DBpedia is exploited. As defined, the method is dataset and domain specific. Aemoo [20] is another Linked Data exploratory system which provides a summary of knowledge about entities. DBpedia is used as primary source of knowledge and it can be utilized for many domains. In [19] the authors describe Discovery Hub, an exploratory search engine which recommends resources from the DBpedia namespace. It uses an adapted version of the spreading activation algorithm over typed graphs [84]. The method exploits the spreading activation method in order to find semantic relatedness between items from different domains. Thus, DiscoveryHub can be classified as multi-domain exploratory search system. The system exploits only a small portion of the available information in DBpedia – triples with properties dcterms:subject and rdf:type, and the DBpedia Pagelinks partition. A content-based and context-aware approach which adopts semantics is presented in [85]. The approach, named Contextual eVSM, is based on distributional models and entity linking techniques. Entity linking is executed in order to extract concepts mentioned in text and link them to an LOD dataset in order to retrieve additional structured information for modeling user preferences. In [86] the authors propose a topic recommender. The approach is based on DBpedia and it utilizes the proximity in the DBpedia knowledge graph. The goal is to enable discovery of relevant but unexpected and potentially unknown topics for the user.

2.2.3.1 Research Gaps

The recent efforts towards using Linked Data in the context of recommender systems have shown promising results. Table 2.4 presents an overview of the most relevant Linked Data Recommenders and Exploratory Search Systems. According to the review, we have identified the following research gaps (RG):

- **RG3.1: Domain specific adoption.** Being able to easily adapt a method to a specific domain is highly important. While most methods have been designed and
validated for a particular domain (e.g. movies or music), there are many other domains which can benefit from Linked Data recommenders. Vast majority of the existing Linked Data recommender approaches usually require human expertise from the particular domain [21, 22, 23, 24, 25, 26, 27, 28, 29], or manual pre-processing of the domain dataset [30] in order to adapt them. Nevertheless, there are a few multi-domain approaches [19, 20] which can deal with several domains and provide cross-domain recommendations.

- **RG3.2: Dataset consumption and adaptation.** The LOD cloud provides access to many datasets which can provide relevant information for Linked Data Recommenders. Nevertheless, majority of the Linked Data recommenders are applied on a single LOD dataset and consume just a subset of the available information from that dataset. Also, most of the methods are exclusively designed for a specific dataset and their adoption for other datasets is impossible or requires significant human effort. Particular method which addresses the “dataset adaptation” issue is RecSPARQL [26]. It enables users to develop customized recommender over an arbitrary RDF graph. However, a human intervention is needed in order to define the list of relevant features which are utilized when evaluating recommendations.

- **RG3.3: Exploitation of available information.** An LOD dataset consists of instance data (A-Box) as well as a schema described with an ontology (T-Box). Both, the instance data and the schema information, are important and can contribute to a recommender system. However, majority of Linked Data recommenders only consider the instance data, while the schema has been only partially considered in [31, 23, 24, 26].

- **RG3.4: Real-time recommendations.** A recommender system should also consider the dynamics of the dataset in terms of new items. However, due to different reasons, such as the high complexity of the underlying methods or the low availability of the SPARQL endpoints, most Linked Data recommenders perform offline pre-processing of the datasets. Only DiscoveryHub [19] and RecSPARQL [26] allow on-the-fly generation of recommendations. On-the-fly processing of the data is also important in order to ensure most up-to-date information.

**Progress beyond the state-of-the-art.** In order to address the aforementioned research gaps, we have developed a novel configurable method for personalized retrieval of Linked Data. The method can be easily adapted to a dataset from any domain and make use of it (RG3.1, RG3.2). It exploits a complete dataset, including the instance and schema information (RG3.3) and does not require any manual pre-processing of the dataset. A dataset can be provided as a dump, but fresh data can be also retrieved via a SPARQL endpoint and generate on-the-fly recommendations (RG3.4). The method, its algorithms, experimental evaluations and validation scenarios are described in Chapter 5.
Table 2.4: Summary of Linked Data Recommenders and Exploratory Search Systems.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Type</th>
<th>Relation to Linked Data</th>
<th>Method</th>
<th>Domain adoption</th>
<th>Dataset consumption</th>
<th>Dataset adoption</th>
<th>Real-time recommendations</th>
<th>Exploitation of Ontological Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbrec [30]</td>
<td>Recommender</td>
<td>Recommender on Linked Data</td>
<td>Semantic Distance</td>
<td>Manual pre-processing</td>
<td>Subset from DBpedia</td>
<td>Dataset specific</td>
<td>Offline</td>
<td>Instance Data</td>
</tr>
<tr>
<td>Heitmann and Hayes [21]</td>
<td>Recommender</td>
<td>Linked Data exploration</td>
<td>Collaborative Filtering</td>
<td>Multi-domain</td>
<td>Subset from DBpedia and DBTune</td>
<td>Dataset specific</td>
<td>Offline</td>
<td>Instance Data</td>
</tr>
<tr>
<td>Discovery Hub [19]</td>
<td>Recommender</td>
<td>Linked Data exploration</td>
<td>Spreading Activation</td>
<td>Multi-domain</td>
<td>Subset from DBpedia</td>
<td>Dataset specific</td>
<td>On-the-fly</td>
<td>Instance Data</td>
</tr>
<tr>
<td>Aemoo [20]</td>
<td>ESS</td>
<td>Linked Data for information search</td>
<td>EKPs</td>
<td>Specific domain</td>
<td>Subset from DBpedia</td>
<td>Dataset specific</td>
<td>Offline</td>
<td>Instance Data</td>
</tr>
<tr>
<td>LED [31]</td>
<td>Recommender</td>
<td>Recommender fed by Linked Data</td>
<td>DBpedia Ranker</td>
<td>Requires domain expertise</td>
<td>Subset from IT and EN DBpedia</td>
<td>Dataset specific</td>
<td>Offline</td>
<td>Instance Data</td>
</tr>
<tr>
<td>Cinemappy [21, 22]</td>
<td>Recommender</td>
<td>Recommender fed by Linked Data</td>
<td>Content Based Filtering</td>
<td>Requires domain expertise</td>
<td>Subset from Freebase</td>
<td>Dataset specific</td>
<td>On-the-fly</td>
<td>Instance Data</td>
</tr>
<tr>
<td>Fernández-Tobías et al. [23]</td>
<td>Recommender</td>
<td>Recommender fed by Linked Data</td>
<td>Network Weight</td>
<td>Requires domain expertise</td>
<td>Custom RDF dataset (MovieLens)</td>
<td>Dataset specific</td>
<td>Offline</td>
<td>Instance Data</td>
</tr>
<tr>
<td>Lommatzsch et al. [24]</td>
<td>Recommender</td>
<td>Recommender fed by Linked Data</td>
<td>Graph-based link prediction</td>
<td>Requires domain expertise</td>
<td>subset of DBpedia and EventMedia</td>
<td>Dataset specific</td>
<td>Offline</td>
<td>Instance Data</td>
</tr>
<tr>
<td>RecSPARQL [25]</td>
<td>Recommender</td>
<td>Recommender fed by Linked Data</td>
<td>Collaborative and Content-based Filtering</td>
<td>Requires domain expertise</td>
<td>subset of DBpedia</td>
<td>Dataset specific</td>
<td>Offline</td>
<td>Instance Data</td>
</tr>
<tr>
<td>Khrouf and Troncy [26]</td>
<td>Recommender</td>
<td>Recommender fed by Linked Data</td>
<td>Collaborative and Content-based Filtering</td>
<td>Requires domain expertise</td>
<td>subset of DBpedia</td>
<td>Dataset specific</td>
<td>Offline</td>
<td>Instance Data</td>
</tr>
<tr>
<td>Rowe [28, 29]</td>
<td>Recommender</td>
<td>Recommender fed by Linked Data</td>
<td>Matrix Factorization</td>
<td>Requires domain expertise</td>
<td>subset of DBpedia</td>
<td>Dataset specific</td>
<td>Offline</td>
<td>Instance Data</td>
</tr>
</tbody>
</table>
Acquisition and Semantization of Web API Descriptions

In the last decade, the number of published datasets as part of the Linked Open Data has significantly increased. From only 294 datasets in September 2011, the number of published datasets has raised to 1,146 in January 2017\(^1\). These datasets cover different domains, such as life sciences, government, geography and linguistics. Nevertheless, specific domains, such as the Web services domain, are not covered, or their coverage is very minimal. There are several benefits in having a Linked Data dataset with Web service descriptions. The service descriptions are contextualized, they can be referenced, re-used and combined. This, in turn, enables analysis of the recent trends in the Web service ecosystem, evaluation of the Web services popularity, comparison of Web services, and automatic Web services composition. In other words, developers can efficiently discover and select Web service, while Web service providers and analysts can get better insight and analysis on the overall Web service ecosystem.

This chapter describes our work within the first activity of the overarching process, the knowledge acquisition and semantization. In particular, we describe our work on knowledge acquisition, semantization and Linked Data publishing in the context of Web services. The results from this work, the dataset and the ontology, are exploited within our work on personalized knowledge retrieval where we develop a method for personalized retrieval of Linked Data (see Chapter 5). In our work, as source of information for the dataset we considered ProgrammableWeb, the largest Web API and mashup repository. We first describe the available information and how the data was collected (see Section 3.2). For modelling relevant Web APIs information, we have developed a light-weight ontology called the Linked Web APIs ontology (see Section 3.3). The ontology builds on top of existing and well established ontologies and appropriately extends them. The ontology is used to model the collected Web API information and generate the Linked Web APIs dataset (see Section 3.4). We also describe the coverage, availability and the maintenance and sustainability plans for the dataset and the ontology. We interlinked the dataset with

\(^1\)http://lod-cloud.net/
3. Acquisition and Semantization of Web API Descriptions

several other LOD datasets and published according to the Linked Data principles (see Section 3.5). The dataset and the ontology have been also evaluated for their quality (see Section 3.6). In order to illustrate the potential of the dataset, we present several use cases on personalized Web API retrieval, automated Web API discovery, composition and orchestration, and temporal analysis of the Web API ecosystem (see Section 3.7.1). We also executed a survey to evaluate the usefulness of the dataset for different audiences such as Web API consumers, providers and analysts (see Section 3.7.2). Finally, we discuss our future work plans with regards to related vocabularies and potential data sources in Section 3.8 and summarize our work in Section 3.9.

Our main contributions presented in this chapter are threefold.

- A **Linked Data dataset with semantic Web API descriptions named Linked Web APIs**, which is largest of its kind. We provide evidence that Web Service descriptions enhanced with semantics enable more advanced analysis of the Web service ecosystem that was not possible before (see Section 3.7.1) and more accurate retrieval of Web services compared to the traditional non-semantic based recommendation mechanisms (see Section 5.3.3).
- A **light-weight ontology for modelling relevant Web API information**, which enables efficiently to capture all relevant Web service information that is not possible with the existing semantic models (see Section 3.3).
- A **survey on the usefulness of the dataset** which provides an evidence and ascertains the added value and the degree of achievement (see Section 3.7.2).

The work described in this chapter is guided by the following set of research questions on which we provide answers:

- **RQ1.1**: “What are the benefits of a dataset with semantic Web service descriptions?” We identify the benefits from a dataset with semantic Web service descriptions and provide support for our work on the dataset (see Section 3.7.1 and Section 3.7.2).
- **RQ1.2**: “To what extent do semantics improve the accuracy of the process of Web service retrieval?” We provide evidence that semantics improve the accuracy of recommendations compared to the non-semantic based recommendation mechanisms (see Section 5.3.3)\(^2\).
- **RQ1.3**: “How can we improve the efficiency of modeling relevant Web service information?” We review existing semantic Web service models and based on the findings, we develop an integrated light-weight ontology which can efficiently capture all relevant Web service information (see Section 3.8.1 and Section 3.3).
- **RQ1.4**: “To what extent do different types of Web service users find a dataset with semantic Web service descriptions useful?”

\(^2\)Please note that RQ1.2 is related to our work on knowledge acquisition and semantization and the work behind the research questions is described in Section 5.3.3.
3.1. Introduction

We evaluate the benefit (usefulness) from the created dataset for different types of Web service users (see Section 3.7.2).

The work described in this chapter is based on a Semantic Web Journal paper [A.1] and an ISWC 2012 research paper [A.3]. The research paper “Personalised Graph-based Selection of Web APIs” was nominated for best research paper and best student paper at the ISWC 2012 conference.

3.1 Introduction

Web APIs enjoy a significant increase in popularity and usage in the last decade. They have become the first-class citizens on the Web and the core functionality of any Web application for exposing functionalities and data. Targeting the developers audience, they lower the entry barriers for accessing valuable enterprise data and functionalities. Back in late 2008, ProgrammableWeb.com, the largest Web API and mashup directory, reported only 1,000 Web APIs. This count increased to 5,000 APIs in Feb 2014 and over 13,000 APIs in June 2015. Nevertheless, most Web APIs, including those listed at ProgrammableWeb, lack machine-readable semantic descriptions. As a consequence, their discovery, sharing, integration, and assessment of their quality and consumption is limited.

There are several benefits in having these Web API descriptions provided as Linked Data. The Web APIs information is linked, therefore API consumers can effectively discover new Web APIs. Also, the developers can benefit from sophisticated discovery and selection queries for discovery and selection of APIs, while Web API providers can execute queries to get better insight information on the Web APIs ecosystem, the popularity of their APIs or the current trends.

In order to achieve these goals, we have developed the Linked Web APIs dataset, an RDF dataset with semantic Web API descriptions. It provides information about Web APIs, mashups (i.e. Web API compositions), and mashup developers. It provides semantic descriptions for 11,339 Web APIs, 7,415 mashups and 7,717 developer profiles, which make it the largest dataset from the Web APIs domain. The primary source for the dataset is the ProgrammableWeb.com directory, which acts as central repository for Web API descriptions. The dataset captures the provenance, temporal, technical, functional, and non-functional aspects. For modelling the captured information we developed a lightweight ontology. In order to conform to the Linked Data principles, we have also linked the dataset with four central LOD datasets: DBpedia, Freebase, LinkedGeoData and GeoNames.
3. Acquisition and Semantization of Web API Descriptions

3.2 The Data Source

In our work, we have considered ProgrammableWeb as a primary source of information for creating the dataset. It adopts characteristics of a social Web platform where Web API providers can publish and share information about offered Web APIs and consequently increase their visibility. The API directory also allows developers to find appropriate APIs for their projects, or they can learn from showcases of existing mashup applications.

The implemented knowledge extraction process consists of four steps: (1) parsing and extraction of valuable information from pages describing Web APIs, mashups and developers, (2) pre-processing, cleanup and consolidation of information, (3) linking with LOD resources, and (4) lifting in RDF and publishing the data as Linked Data.

An example of a Web page which describes a Web API is the one which describes the Twitter API\(^{10}\). For each Web API we extracted its title, short summary describing its functionalities, assigned tags and categories, technical information, such as supported formats and protocols, as well as non-functional properties such as its homepage, usage limits, usage fees, security, etc. Similarly, for each mashup we extracted its title, short free-text description of its functionalities, assigned tags and the homepage. From each page which describes a developer, we extracted its username, homepage and short bio. The city and country of residence, its given and family name and the gender were extracted only if these information were publicly available.

We also captured the relationships between the Web APIs, mashups and developers. In other words, for each mashup we extracted the list of Web APIs which were used by the mashup and also the list of mashups created by each developer. The dataset also captures the temporal aspects - the creation time of the Web APIs, mashups and the time a user registered his profile.

In order to collect the data, we have implemented a script which systematically browses and parses relevant pages. The parsing mechanism has been implemented using the jsoup Java HTML parser\(^{11}\). For the crawler, we used proper etiquette and we configured the crawl delay to one page every four seconds.

3.3 The Ontology

The Linked Web APIs ontology\(^{12}\) is a minimal model that captures the most relevant information related to Web APIs and mashups. Since existing semantic Web service solutions do not completely capture all available Web API information, we have decided to develop an ontology which builds on top of the existing and well established ontologies and appropriately extend them. See Section 3.8.1 for more information on how our ontology compares to the related ontologies.

\(^{10}\)http://www.programmableweb.com/api/twitter  
\(^{11}\)http://jsoup.org/  
\(^{12}\)http://linked-web-apis.fit.cvut.cz/ns/core/index.html
3.3. The Ontology

The selection of the ontologies for integration was driven by the following four crucial requirements:

- **Provenance:** It is important to keep information about *Who* (developers) created *What* (mashups) and *How* (using which APIs). In addition, information about API providers need to be captured as well.

- **Functional and Non-functional Properties:** *What* functionalities a Web API or mashup offers is more than important, as well as their usage limits and fees, supported security and authentication mechanisms.

- **Technical Properties:** Information about the supported protocols, formats and the Web API endpoint location is as important, as it allows a Web API consumer to search only for APIs with preferred technical capabilities.

- **Temporal Information:** *When* a mashup or Web API was created is a valuable information as well. For example, to analyze the recent trends in the API ecosystem or to discover the most recent Web APIs or mashups.

Figure 3.1 shows the overall Linked Web APIs ontology. The ontology contains three central classes: *iso:WebAPI* – describes Web APIs, *iso:Mashup* – describes mashup compositions, which utilize one or more Web API, and *iso:Agent* – represent all kinds of entities involved in the creation and/or consumption of Web APIs and mashups.
In order to capture the provenance information, the Linked Web APIs ontology integrates the PROV-O ontology\(^\text{13}\) by incorporating the classes \textit{prov:Entity}, \textit{prov:Activity} and \textit{prov:Agent}, and their related properties. The \textit{prov:Entity} class serves as super-class of \textit{iso:WebAPI} and \textit{iso:Mashup} classes. The activities convey information about the consumption process of the Web APIs and the generation of mashups by the agents. Note that an activity can also refer to the action of the creation of an API documentation (i.e., a ProgrammableWeb entry) and this can be modeled by associating an action with an instance of the \textit{hydra:ApiDocumentation} class from the Hydra\(^\text{14}\) vocabulary. We introduce the \textit{iso:usedAPI} property which refines the semantics of the \textit{prov:used} property so it can be used to explicitly identify usage of a Web API in a mashup creation. The temporal information about the creation time of a mashup or Web API is expressed using the \textit{prov:generatedAtTime} property. It represents the time when the API documentation has been generated within the snapshot. Since this is the first official snapshot of the dataset, the value of this property indicates the first version of each resource. Any changes in the following snapshots will be captured with the \textit{dcterms:modified} property. Also note that in the most cases the \textit{prov:generatedAtTime} is the time when the API has been registered at ProgrammableWeb. For example, the Google Maps API, the Twitter API and the YouTube API, are between the most popular APIs at ProgrammableWeb, and the provided time information are justifiable; May 5th, 2005 for the Google Maps API, August 12th, 2006 for the Twitter API, and August 6th, 2006 for the YouTube API.

For the functional (tags and categories) and non-functional (formats and protocols) properties of the Web APIs and mashups, we introduce new classes in our namespace. The ontology also integrates the \textit{wl:NonFunctionalParameter} class from the WSMO-lite ontology\(^\text{15}\) [7], which was developed by the Semantic Web Services community, to explicitly identify non-functional properties. The Minimal Service Model (MSM)\(^\text{16}\) ontology, which was initially defined for the hRESTS microformat [8], is also considered. The class \textit{msm:Service} is integrated as super-class of the \textit{iso:WebAPI}. This allows us to attach additional Web API information, such as operations, inputs and outputs, which is relevant for the execution of Web APIs. General metadata information such as a Web API and mashup title, or their short textual description are described using the Dublin Core vocabulary\(^\text{17}\).

\(^{13}\)http://www.w3.org/TR/prov-o/
\(^{14}\)http://www.hydra-cg.com/spec/latest/core/
\(^{15}\)http://www.wsmo.org/ns/wsmo-lite/
\(^{16}\)http://iserve.kmi.open.ac.uk/ns/msm#
\(^{17}\)http://purl.org/dc/terms/
3.4 The Linked Web APIs Dataset - Coverage, Availability and Maintenance

3.4.1 Coverage and Availability

The Linked Web APIs dataset is the first of its kind with descriptions for more than 11,339 Web APIs, 7,415 mashups and 7,717 mashup creators. Overall, the dataset contains over 550K RDF triples. See the appendix \(^ \text{18} \) for an example of a resource instance. For all the resources we mint URIs in our own namespace (http://linked-web-apis.fit.cvut.cz/resource/{name}). The name part from the URIs is a normalized form of the resource label, which is lowercased and each space is replaced with an underscore sign. Since it is possible that two different resources have the same name (e.g., the label XML can occur as a tag and also as a format), we attach its type as a suffix to the URI. For example, _tag for tags or _api for Web API URIs. The URI http://linked-web-apis.fit.cvut.cz/resource/google-maps_api is an example of a URI minted for the Google Maps API. A similar approach is employed by DBpedia and Wikipedia\(^ \text{19} \) to distinguish pages which have the same title. For example, /resource/Food_(band) for a page describing the musical band “Food” and /resource/Food_(film) for a page describing the movie with the same name. It would be possible to use an alternative schema for the URIs, where the resource type is encoded as a path component (e.g. /resource/apis/google-maps). Which URI schema is more appropriate is a debatable question. Nevertheless, both approaches are valid and serve their purpose.

All the URIs are dereferenceable and served according to the Linked Data principles in RDF/XML and Turtle format. The dataset is available via a Virtuoso SPARQL endpoint and as an RDF dump. The landing page for the dataset is http://linked-web-apis.fit.cvut.cz/ and it provides information about the latest news, releases and changes. Technical details about the dataset are listed in Table 3.1.

Table 3.1: Details of the Linked Web APIs dataset.

<table>
<thead>
<tr>
<th>Name</th>
<th>Linked Web APIs dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
<td><a href="http://linked-web-apis.fit.cvut.cz/">http://linked-web-apis.fit.cvut.cz/</a></td>
</tr>
<tr>
<td>Endpoint</td>
<td><a href="http://linked-web-apis.fit.cvut.cz/sparql">http://linked-web-apis.fit.cvut.cz/sparql</a></td>
</tr>
<tr>
<td>Ontology</td>
<td><a href="http://linked-web-apis.fit.cvut.cz/ns/core#">http://linked-web-apis.fit.cvut.cz/ns/core#</a></td>
</tr>
<tr>
<td>Version</td>
<td>0.1</td>
</tr>
<tr>
<td>Ver. Date</td>
<td>05.08.2015</td>
</tr>
<tr>
<td>License</td>
<td>Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)</td>
</tr>
<tr>
<td>Datahub</td>
<td><a href="https://datahub.io/dataset/linked-web-apis">https://datahub.io/dataset/linked-web-apis</a></td>
</tr>
</tbody>
</table>

\(^ \text{18} \text{app.twitter-api} \)

\(^ \text{19} \text{https://en.wikipedia.org/wiki/Wikipedia:Article_titles#Disambiguation} \)
Currently, we employ the same versioning approach as the one used by DBpedia - versioning at the dataset level. Nevertheless, versioning at the resource level will be considered in the near future. Versioning at the resource level would be appropriate when integrating the APIs.io repository (see Section 3.8.2 for more details), since the API versioning information is explicitly present.

### 3.4.2 Maintenance and sustainability

The computer center of the Czech Technical University in Prague kindly provided us with persistent web space for the publication of the dataset and the ontology. This will guarantee persistent URI identifiers for the dataset resources.

The ongoing maintenance of the dataset is carried out at the data level, as well as at the ontology level and its alignment with relevant existing and emerging vocabularies.

Our long-term goal is to establish the Linked Web APIs as a central Linked Data hub for Web API descriptions. To this end, we aim at providing support for various Web API description models (cf. Section 3.8.1) and data sources with relevant Web API information (cf. Section 3.8.2).

It takes over 29 hours (with four seconds crawl delay) to complete the crawling, information extraction and RDF generation process. The process is fully automated and currently it has to be manually triggered. The dataset has been already integrated with DBpedia (via `owl:sameAs` links; see Section 3.5) and we plan to synchronize the Linked Web APIs dataset generation with the DBpedia releases and generate the dumps on bi-annual basis. Since new APIs are published every day, we also plan to provide a live extraction service which pulls updates in real-time and updates the triple store.

### 3.5 Dataset Linking

In order to assure maximal reusability and integrability, we linked the dataset with four central LOD datasets. Two multi-domain datasets, DBpedia and Freebase, and two geographical datasets, GeoNames and LinkedGeoData. From the information we linked the Web APIs supported data formats, supported protocols, developers’ city and country of residence. Since GeoNames and LinkedGeoData are geographical datasets, only users’ city and country of residence were linked to those datasets. DBpedia and Freebase are multi-domain datasets and therefore we linked all information to these datasets. The links to DBpedia, and respectively to Freebase, were generated following the most-frequent-sense based approach which is used as entity linking method in the Entityclassifier.eu NER system [A.6] (see Chapter 4). The linking to LinkedGeoData was governed by the intuition that the names of the cities and countries in our dataset have same names in the LinkedGeoData dataset. The approach was supported by SPARQL queries which retrieve resources with a given label. Following this linking methodology we generated 1,440

\[\text{http://apis.io/}\]
3.6. Quality

links out of which 722 are DBpedia links, 299 Freebase links, 326 GeoNames links and 93 LinkedGeoData links. Table 3.2 provides more information about the linking.

Table 3.2: Number of linked resources per type and dataset.

<table>
<thead>
<tr>
<th></th>
<th>DBpedia</th>
<th>Freebase</th>
<th>LGD</th>
<th>GeoNames</th>
</tr>
</thead>
<tbody>
<tr>
<td>Formats</td>
<td>283</td>
<td>208</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td>Protocols</td>
<td>123</td>
<td>91</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td>Cities</td>
<td>263</td>
<td>/</td>
<td>47</td>
<td>276</td>
</tr>
<tr>
<td>Countries</td>
<td>53</td>
<td>/</td>
<td>46</td>
<td>50</td>
</tr>
<tr>
<td>Total</td>
<td>722</td>
<td>299</td>
<td>93</td>
<td>326</td>
</tr>
</tbody>
</table>

We opted for these linking approaches, since we have the tooling in place and they served their purpose.

It is important to note that our dataset has also received in-links\(^{21}\) from DBpedia, the most prominent LOD dataset. The links have been accepted and picked up with the DBpedia release from October 2015.

3.6 Quality

According to the 5-star classification system [2], defined by Tim Berners-Lee, the Linked Web APIs dataset classifies as a five-star dataset. The five stars are credited for the open license, availability in a machine-readable format, use of open standards, use of URIs for identification, and the links to the other LOD datasets.

3.6.1 Dataset Quality

Zaveri et al. [87] provide a list of indicators for evaluation of the intrinsic quality of Linked Data datasets. According to the guidelines described by Zaveri et al., we evaluated the datasets for its \textit{semantic accuracy}. We therefore checked whether the data values correctly represent the real world facts. To this end, we have randomly created a set of 100 triples and manually checked their validity. Only two triples representing tags have been spotted as invalid. Note that no invalid triples were spotted for the provenance, technical and non-functional information.

The sample size was determined by considering a confidence level of 95% and confidence interval of 10. In our future work, we plan to extend our evaluation with higher confidence level and lower interval.

3.6.2 Vocabulary Quality

According to the 5-star vocabulary classification [88], the Linked Web APIs ontology credits four out of five stars: for the machine and human-readable information about the vocabu-

\(^{21}\)http://downloads.dbpedia.org/2015-10/links/
lary (2 stars), it is linked to other vocabularies such as WSMO-lite and PROV-O (3 stars) and for the provided metadata information for the vocabulary (4 stars). The fifth star is credited for vocabularies which have been referenced by other vocabularies. However, the Linked Web APIs vocabulary has not been yet used and referenced by other vocabularies.

3.6.3 Known Shortcomings

The information extraction process is not entirely flawless due to its dependency on the HTML structure. From early 2012, when we created the initial snapshot of the dataset, until early 2016, the HTML has changed only two times, which is approximately every two years. Nevertheless, we are also considering other potential data sources, which will soon be integrated as part of the Linked Web APIs dataset. APIs.io and mashape marketplace\(^{22}\) are the two most prominent data sources (see Section 3.8.2 for the discussion about datasources). We are currently working on an integration of APIs.io and soon it will be part of the Linked Web APIs dataset.

As for the ontology, some properties, such as “usageFees” and “usageLimits”, are currently modeled as plain literals. The main reason for such a decision was the diversity of the possible values of these properties in the data. Very often these properties are expressed in a natural language, thus its modeling is a challenging task. In the future, if a data source provides data of a greater quality for these properties, we will appropriately extend the ontology.

3.7 Usefulness of the Dataset

3.7.1 Use Cases

The availability of a dataset with Web APIs descriptions in RDF can support various use cases, including, but not limited to personalised Web API provisioning, API ecosystem analysis and automated processing of Web API descriptions. In this section, we describe selected use cases and existing applications of the Linked Web APIs dataset.

Use case 1: Personalised Recommendations. The Linked Web APIs dataset contains links between the mashup and the developer resources, which can be used as a pertinent source of information for developing Web API recommendation methods. As an example, a user, who has already picked a Web API for his/her mashup, could search for other compatible Web APIs. Such a scenario can be supported with the SPARQL query from Listing 3.1 which returns the top 5 most used Web APIs.

\(^{22}\)https://market.mashape.com/
3.7. Usefulness of the Dataset

In a different scenario, a developer could customize the search query to narrow down the results to Web APIs which support a particular data format (e.g., JSON) or APIs from a specific category (e.g., social, government, etc.).

In the context of personalised recommendations, the dataset has been recently employed in several works around personalised recommendation of Linked Data resources (see Chapter 5 and the associated paper [A.2]) and recommendation of Web APIs, described in [A.3]. These works develop methods on top of the dataset and accommodate the user preferences by analyzing their history. The method described in Chapter 5 recommends resources of interest for users with similar tastes. The method presented in Chapter 5 and the one in [A.3] focus on developing graph based algorithms on top of the Linked Web APIs dataset which utilize the provisioning information (who developed what), functional properties (tags and categories) and temporal information (when a mashup or an API was developed). The target audience in both methods are ultimately API consumers.

Use case 2: Support for Automated API Discovery, Composition and Orchestration. There are semantic models which provide mechanisms for automated Web service discovery, composition and orchestration. SADI [9] defines a mechanism for fully automated processing and integration of Web services. In such scenarios, the Linked Web APIs dataset can be used as a relevant source for discovery of Web APIs and their use in a composition workflows. Assuming a user composer already picked her/his favorite API(s), with a query which is similar to the one in Listing 3.1, then he/she can retrieve a list of additional relevant APIs. These candidate APIs can be further validated and added to the composition workflows.

Use case 3: Temporal Analysis. The dataset also captures the temporal aspect, i.e., the time when a mashup or a Web API was developed. These kind of information can help Web APIs providers to get better insights about the recent developments and study the consumption of a Web API, or the whole Web API ecosystem over a period of time. The benefits from having temporal information can be illustrated with the SPARQL query from Listing 3.2.
3. Acquisition and Semantization of Web API Descriptions

```
SELECT COUNT(?mashup) as ?count
WHERE {
  ?mashup prov:wasGeneratedBy ?activity.
  ?activity lso:usedAPI ls:google-maps_api .
  ?mashup prov:generatedAtTime ?date .
  FILTER (?date >= "2013-01-01"^^xsd:dateTime
     && ?date < "2014-01-01"^^xsd:dateTime)
}
```

Listing 3.2: Number of mashups utilizing the Google Maps API in 2013.

The SPARQL query in the listing gives information about the total number of mashups which utilized the Google Maps API in 2013. Figure 3.2 visualizes the results of the analysis for three popular APIs and their utilization over a period of time.

![API Utilization Graph](image)

Figure 3.2: Web API utilization over time.

The Web API provider might be also interested to find out in what kind of mashups their API was used. An answer to such a question can be answered with the SPARQL query in Listing 3.3.

```
SELECT ?category (COUNT(?category) as ?count)
WHERE {
  ?mashup prov:wasGeneratedBy ?activity.
  ?activity lso:usedAPI ls:google-maps_api .
} ORDER BY DESC(?count)
```

Listing 3.3: The number of mashup categories the Google Maps API was used.

Further, a Web API analyst might be interested in the latest trends in the API ecosystem. Questions such as “What protocols and formats are the most supported by the APIs?” or “Which domains provided the most APIs in 2013?” are likely to occur. Using the
3.7. Usefulness of the Dataset

SPARQL query in Listing 3.4 we can get the top 5 most popular protocols in 2013. Figure 3.3 illustrates the popularity of the HTTP APIs and SOAP based services, for a period of ten years.

```
SELECT ?protocol (COUNT(?api) as ?count)
WHERE {
    ?api rdf:type lso:WebAPI .
    ?api prov:generatedAtTime ?date .
    FILTER (?date >= "2013-01-01"^^xsd:dateTime
        && ?date < "2014-01-01"^^xsd:dateTime)
} ORDER BY DESC(?count)
LIMIT 5
```

Listing 3.4: The most popular API protocols in 2013.

![Popular APIs and SOAP services over time](image)

Figure 3.3: Popularity of HTTP APIs and SOAP based services over time.

An answer to the question “Which domains provided most APIs in 2013?” can be answered with the SPARQL query in Listing 3.5.

```
SELECT ?category (COUNT(?api) as ?count)
WHERE {
    ?api rdf:type lso:WebAPI .
    ?api prov:generatedAtTime ?date .
    FILTER (?date > "2012-01-01"^^xsd:dateTime
        && ?date < "2013-01-01"^^xsd:dateTime)
} ORDER BY DESC(?count)
LIMIT 5
```

Listing 3.5: The most popular API categories in 2013.
The results show that the most popular API category is “tools”, followed by the “science”, “internet”, “enterprise” and “financial” categories. It is interesting that the “financial” and the “enterprise” categories are among the top five most popular API categories, which indicates that APIs are already understood as a relevant technology also by other domains than the internet and the social networks.

A more in-depth analysis using the Linked Web APIs dataset has been conducted in [A.24]. In particular, the dataset has been used as a reference dataset for link discovery in RDF graphs.

3.7.2 Survey on the Usefulness and Potential of the Dataset

In order to evaluate the potential and the usefulness of the dataset we have executed a survey\(^{23}\). The survey targeted people who consume, develop and/or provide Web APIs. In the survey participated 29 people and all of the participants stated that they have searched or used an API, while 19 stated that they also provide an API. The results from the survey show that most of the developers have difficulties while searching an API - 3.4% find it very hard, 41.4% hard, and 27.6% somewhat hard. In addition, the majority of the participants welcome a central API repository - 34.5% find it very helpful, 37.9% helpful, 20.7% somewhat helpful, and 6.9% little helpful. In the survey, we have asked the participants to indicate the usefulness of the Linked Web APIs dataset from the perspective of a Web API consumer and provider. The results (cf. Figure 3.4) show that both, the consumers and the providers find the dataset useful. The results also show that the dataset appears to be more useful for the API consumers than for the API providers.

\(^{23}\)The survey document can be found in the appendix A.1.
3.7. Usefulness of the Dataset

Based search in service directories such as ProgrammableWeb, or iii) by asking other developers for an advice. The results are as follows:

- Using search engines such as Google (27 users); 18% very useful, 54% useful and 28% somewhat useful.
- Running keyword-based search in service directories such as ProgrammableWeb (6 users); 33% very useful, useful and somewhat useful.
- Asking other developers for an advice (21 users); 19% very useful, 48% useful and 33% somewhat useful.

Further, in order to evaluate new potential third-party uses of the dataset, we have also asked the participants if they will consider using the dataset in the near future (cf. Figure 3.5). As shown in Figure 3.5, consumers have shown more interest in using the dataset than the providers.

Figure 3.5: Will consumers (left) and providers (right) consider the dataset in the near future.

Finally, in the survey we have evaluated the usefulness of the dataset on several use cases. The results are as follows:

- 86% – Find and select relevant APIs.
- 86% – Increase the visibility of the APIs.
- 62% – Evaluate the recent trends in the API ecosystem.
- 59% – Compare APIs to others.
- 52% – Automated composition of Web APIs.
- 38% – Track the popularity of the Web APIs.

It can be observed that our goals for the dataset are well aligned with the possible use cases, as seen by the Web API consumers and providers. The results from the survey are also available online.\(^{24}\)

\(^{24}\)Results from the survey: https://dx.doi.org/10.6084/m9.figshare.3459044.v2

67
In overall, the results from the survey confirm the usefulness and the potential of the Linked Web APIs dataset.

3.8 Discussion and Future Work

3.8.1 Relation to Existing Ontologies

In the last decade, several ontologies for formal description of Web services have been developed. OWL-S [5], WSMO [6], SAWSDL [25], WSMO-lite [7], SADI [9], Hydra [26], hRESTS [8] and the MSM [27] ontology, define models for semantic Web service descriptions. However, these semantic Web service models are too complex, they do not completely capture the available Web service information or they address architectural models which are nowadays not prevalent on the Web. OWL-S is one of the first, and quite complex model for use from the perspective of an average service developer [10]. It captures functional, non-functional, behavioral and technical properties, however, it does not capture provenance nor temporal information. Similarly, although the WSMO ontology is highly expressive, it does not capture the temporal information and it requires significant manual effort to use it in practice [11]. Further, WSMO-lite does not capture provenance and the temporal information, and at the same time it addresses WSDL services, which are nowadays not prevalent on the Web [12]. Similarly, SAWSDL is also targeting WSDL/SOAP services which are nowadays not prevalent on the Web. hRESTS defines a poshformat, which can be used to annotated HTML pages describing Web APIs. However, hRESTS does not consider the temporal and provenance information, and also, since it is a de facto a microformat with no namespace support, it faces with scalability issues. SADI has been developed exclusively for the bioinformatics domain and it can be used to capture functional and technical information, but there is lack of support for non-functional, behavioral, provenance and temporal information. Hydra is one of the latest development in vocabularies for hypermedia driven Web APIs. It primarily defines classes and properties for description of the service behavioral, functional, non-functional and technical properties, but provenance and temporal information is out of the scope. See Section 2.2.1.1 for a complete review of the existing semantic Web service models.

In response to the above-mentioned issues, we developed a light-weight ontology on top of several well existing ontologies and appropriately extended them. In comparison to the other ontologies, the Linked Web APIs ontology is a light-weight solution which can capture the available provenance, functional, non-functional, technical and temporal information. It builds on top of the WSMO-lite, hRESTS and the MSM Semantic Web Service models and in the near future we will also provide an alignment for the SADI model. SADI provides mechanism for an automated discovery, composition and orchestration of Web services. Since the Linked Web APIs dataset provides large amount of information

25http://www.w3.org/TR/sawsd1/
26http://www.hydra-cg.com/spec/latest/core/
27http://iserve.kmi.open.ac.uk/ns/msm#
about Web APIs, it can efficiently aid the process of discovery of relevant APIs for SADI composition workflows.

Moreover, there are also non-Semantic Web standards such as WADL\textsuperscript{28} and WSDL\textsuperscript{29}, which define syntactic descriptions for Web services. These syntactic descriptions are of high importance for the process of execution of Web services, individually or combined in service compositions. APIs.json\textsuperscript{30} is another API description format which has recently gained attention by the API community. It is a JSON based format for public deployment of API descriptions. In our future work, we plan to support and integrate these API description formats.

In our future work, we also plan to integrate ontologies such as the SPARQL Service Description\textsuperscript{31} ontology and the DataID\textsuperscript{32} dataset description model [89] which will in turn allow description of SPARQL processing services and corresponding Linked Data datasets. Last but not least we want to evaluate possible alignments of the ontology with tagging vocabularies such as the MUTO\textsuperscript{33} and the SCOT\textsuperscript{34} vocabularies.

### 3.8.2 Additional Data Sources

Currently, the Linked Web APIs dataset is populated with data from the ProgrammableWeb repository. Nevertheless, our ultimate goal is to establish the Linked Web APIs as a central Linked Data hub for Web API descriptions. In order to achieve this goal, we are currently working on enriching the dataset with API descriptions from other data sources. Integration of the API repository APIs.io as part of the Linked Web APIs dataset is a currently ongoing effort. The repository provides over 1,000 API descriptions in the APIs.json\textsuperscript{35} format. The APIs.json descriptions are being deployed in a decentralized manner, at the same domain from which the APIs are available. By integrating the APIs.io repository as part of the Linked Web APIs dataset, developers are going to be able to publish and easily maintain their API descriptions, while at the same time, making theirs descriptions available as Linked Data. In our future work, we will also consider integrating API marketplaces, such as the mashape marketplace\textsuperscript{36}.

We also plan to enrich the dataset with user profiles from traditional social networks. We want to interlink the tags and categories information with relevant datasets from the LOD cloud such as the Wikidata\textsuperscript{37}, Wiktionary\textsuperscript{38} and Dbnary\textsuperscript{39}. Last but not least we\textsuperscript{28}\textsuperscript{29}\textsuperscript{30}\textsuperscript{31}\textsuperscript{32}\textsuperscript{33}\textsuperscript{34}\textsuperscript{35}\textsuperscript{36}\textsuperscript{37}\textsuperscript{38}\textsuperscript{39}
want to explore other applications using the dataset and assess its potential.

3.9 Summary

The growing number of available Web APIs requires new mechanisms to support the process of sharing, discovery, integration and re-use of Web APIs at a large scale. In this chapter, we have presented the Linked Web APIs dataset, the first Linked Data dataset which provides Web API descriptions. The dataset supports i) API consumers-in the process of discovery, selection and use of Web APIs, ii) API providers-in increasing the visibility and tracking the popularity of their Web APIs, and iii) API analysts-in analyzing the API ecosystem. The dataset will also help to raise the awareness about the importance of providing semantic Web API descriptions and publishing them as Linked Data. The dataset has been validated in several recent works [A.3, A.2, A.24] in the context of personalized recommendations and link analysis. Also, on a set of usage scenarios we have shown the potential of the dataset.

This chapter described results from our work on knowledge acquisition and semantiza-
tion in the context of Web services. These results are exploited in our work on personalized knowledge retrieval (see Chapter 5).
Knowledge Extraction and Integration with Salient Linked Entities

Although the amount of published structured information is increasing, still large portion of the information on the Web is available only in an unstructured format. In order to realize the vision of the Semantic Web, there is need to extract and transform this hidden information into a structured knowledge, and integrate it into the Linked Data space.

This chapter describes our work within the second activity of the thesis, the knowledge extraction and integration using salient named entities. In this chapter, we describe the developed general purpose Named Entity Recognition system (Section 4.1) [A.6, A.10], which is supported with methods for identification, classification and linking of salient entities. The system has been used to enrich the dataset with semantic Web descriptions, which is the main results from the first activity of the thesis (Chapter 3), and link particular information from the dataset with other Linked Data datasets (see Section 3.5). Evaluations on several challenges and datasets are also presented [A.8, A.7, A.10]. We also present a validation scenario within the EU project LinkedTV\(^1\) and a case study on concept detection in video transcripts [A.11]. Furthermore, we developed a method for learning and extraction of salient entities (see Section 4.2) [A.5]. The method aims at extraction of entities that play an important role in the story that the document describes. The method is supported with a novel corpus with crowdsourced entity salience annotations. In order to enable evaluation of NER systems, we also developed an evaluation framework for NER (Section 4.3) [A.9, A.12]. We describe the approach for NER evaluation, its components and integrated datasets. Finally, Section 4.4 summarizes our achievements on knowledge extraction and integration via salient linked entities.

Our main contributions are fivefold:

- An open-source NER system, named Entityclassifier.eu, supported with a set of methods for spotting, linking and classification (see Section 4.1), which outperforms the

\(^{1}\)http://www.linkedtv.eu/
other related NER systems in terms of accuracy. These evaluations have been executed on the Czech Traveler dataset (see Section 4.1.5.1).

- **An evidence about the accuracy of the developed methods under different conditions (datasets, domains, focus queries and languages).** We show that the most-frequent-sense approach gives more accurate results for Person focused queries compared to Organization and Geopolitical queries. We also evaluate the performance for different document collections and show that our system gives best results for discussion fora documents than for newswire or web documents (see Section 4.1.5.3 and Section 4.1.5.4),

- **A method for learning entity salience based on local and global set of features,** which gives more precise results when compared to a related state of the art method [1] (see Section 4.2.4).

- **A crowdsourced dataset with entity salience annotations.** A publicly available, complete and manually checked dataset with entity salience annotations (see Section 4.2.3).

- **An evidence about the impact of the individual features on the performance of learning entity salience.** We show that a combined set of local and global features gives better accuracy than each set used individually. We also evaluate the impact of each individual feature (see Section 4.2.4).

- **An evidence about the impact of the incorrect links on the performance of learning entity salience.** We show that incorrect links have low impact on the performance of learning entity salience (see Section 4.2.4).

The work described in this chapter is guided by the following set of research questions on which we provide answers:

- **RQ2.1: “How does the quality of our NER system compare to other systems for different datasets and for different type of focus queries?”** - We investigate how our NER system compares in terms of accuracy when compared to other similar systems on different datasets and different focus queries (PER, ORG, GPE). Such evaluations have been executed on the Czech Traveler dataset (see Section 4.1.5.1) and on datasets at the TAC’13 (see Section 4.1.5.3) and TAC’14 (see Section 4.1.5.4) evaluation challenges.

- **RQ2.2: “How accurate results gives our method for identification of salient named entities compared to other similar methods?”** - We investigate how our method for identification of salient entities performs when compared to the most related state of the art method [1] (see Section 4.2.4, Experiment 2).

- **RQ2.3: “What is the impact of individual and combined local and global set of features on the performance of learning entity salience?”** - We investigate the impact of the individual features on the performance of learning entity salience (see Section 4.2.4, Experiment 3).
4.1. Named Entity Recognition and Linking

- RQ2.4: “How does the quality of the entity links influence the performance of learning entity salience?” - We investigate the impact of the quality of the entity links on the performance of learning entity salience (see Section 4.2.4, Experiment 4).

The work described in this chapter is based on several papers [A.5, A.6, A.8, A.7, A.9, A.10], contributions to book chapters [A.11, A.12] and technical reports [A.15, A.13, A.14].

4.1 Named Entity Recognition and Linking

4.1.1 Introduction

Named Entity Recognition (NER) and Entity Linking (EL) enjoy a significant increase in popularity and usage in the last decade. They have become the core technology for knowledge extraction from unstructured data and its integration with Linked Data. The efforts within the Semantic Web and NLP communities gave birth of several NER and EL systems, such as DBpedia Spotlight [13], AIDA [14], NERD [15], Babelly [16], FOX [17], OpenCalais2 and StandfordNER3. Although there exist a number of sophisticated NER and EL systems, there is still a lack of NER systems that are not heavily based on training data and evaluate the importance of the entities in the document. In order to address these issues, we have developed a NER system which assesses the importance of the entities (i.e. entity salience) and performs unsupervised entity spotting and classification. The entity spotting is implemented with lexico-syntactic patterns, while the entity classification applies text mining techniques on top of Wikipedia in order to extract entity types in real-time. Moreover, with support of Wikipedia and DBpedia, the recognized entities are further linked with these knowledge sources.

The work described in this section is based on several papers [A.6, A.8, A.7, A.9, A.10], a book chapter [A.11] and contributions to several technical reports [A.15, A.13, A.14].

4.1.2 Approach

Our approach on NER and EL can be split into three phases. The first phase is the entity spotting phase, which locates entity mentions in a given text. The entity spotting is followed by the entity linking phase, which assigns link to each entity mention. The assigned link is referring to a resource which describes and uniquely identifies the entity mention. Finally, the entity classification phase assigns type for the spotted and linked entity mention. Below we provide detailed description of the entity spotting, linking and classification phase.

2http://www.opencalais.com/
3https://nlp.stanford.edu/software/CRF-NER.shtml
4. Knowledge Extraction and Integration with Salient Linked Entities

4.1.2.1 Entity Spotting Method

Entity spotting is the first phase of our NER solution. It accepts free text (i.e. unstructured text) and outputs a list of named entities and what we call, common entities. Named entities are identified as proper noun phrases which comprise of one or more tokens of type “proper nouns”. Similarly, common entities are defined as common noun phrases which comprise of one or more tokens of type “common nouns”. Named and common entities are formally defined using manually crafted lexico-syntactic patterns based on a given linguistic information. The patterns as linguistic information utilize Part-Of-Speech (POS) tags. Named entities are defined as NNP+, where NNP is a proper noun. An example of a named entity is “Diego Maradona”. Common entities are recognized with the pattern JJ* NN+, where JJ is an adjective and NN is a noun. An example of a common entity is “footballer” or “hockey player”. Note that both “*” and “+” in JAPE grammar are greedy operators. A simple example of an entity spotting pattern implemented using a JAPE grammar [90] is shown in Listing 4.1.

```
Rule: EntityExtractionRule
Priority: 100

(({Token.category == "NNP"})+)

:entityString

--> :entityString.ne = {name="NNP+ rule"}
```

Listing 4.1: JAPE grammar for spotting entities.

4.1.2.2 Entity Linking Methods

The Entity Linking phase is executed right after the spotting phase. The goal is to link an entity mention with its representation in a given knowledge base. In our work, we explored several approaches for entity linking which we describe below.

Most Frequent Sense (MFS) based linking. The MFS method is a context independent method which does not use the context text around the entity, but it uses only the entity name when performing the linking. In this approach, the entity is linked with the most-frequent-sense entity found in the reference knowledge base. To realize the MFS approach we used the available English Wikipedia Search API and we also used a specialized Lucene index⁴ which extends the Apache Lucene search API. It ranks pages based on the number of backlinks and the Wikipedia articles’ titles. Note that the Wikipedia Search API is build on top of the Lucene index and it offers some more functionalities.

Most Frequent Sense (MFS) based linking enhanced with context. To choose the most relevant entity candidate, this approach combines the most frequent sense approach with the context around the entity. To retrieve the set of potential candidates, we submit

⁴http://www.mediawiki.org/wiki/Extension:Lucene-search
4.1. Named Entity Recognition and Linking

A Lucene search query with the entity name. The top-5 most relevant Wikipedia pages are considered as the potential candidates. Next, we extract the entities (using Entityclassifier.eu) from the paragraph where the entity occurs, and we also extract entities from the corresponding DBpedia abstract for each of the Wikipedia candidate pages. Finally, the entity is linked with the page with the highest number of overlapping entities.

**Explicit Semantic Analysis (ESA) based linking.** This entity linking approach is based on the Explicit Semantic Analysis (ESA) method [91]. In the ESA method, the input text $T$ is represented as a TF-IDF term vector. For each word $w_i$ in the input text the method uses an inverted index to retrieve Wikipedia articles $c_1, \ldots, c_n$ containing $w_i$. The semantic relatedness of the word $w_i$ with concept $c_j$ is computed such that the strength of association between $w_i$ and $c_j$ is multiplied with the TF-IDF weight of $w_i$ in $T$. The relatedness score for any two documents is determined by computing the cosine similarity between the vectors of document-concept semantic relatedness. ESA has a number of follow-up papers describing particularly its applications in various areas of information retrieval, including cross-language information retrieval (cf. [92] for an overview). The use of ESA for disambiguation of a surface form to a Wikipedia URL was proposed in [93].

We employ ESA as follows. First, for a given entity mention, we collect $K$ potential entity candidates. Next, for each entity candidate, we retrieve its description. We use the first paragraph from Wikipedia as description for each entity candidate. Finally, we use the ESA\(^5\) method to compute the similarity between each entity candidate description and the context text around the entity mention. In our experiments (see Section 4.1.5.3), the entity context is a 800 characters log text which consists of the 400 characters preceding and following the entity mention. After computing the similarity between each description and the entity mention context text, the entity with the highest similarity score is considered as correct and it is linked with the entity in the reference knowledge base.

4.1.2.3 Entity Classification Method

Entity classification is an integral part of the most NER systems. It aims at assigning a type to a given entity mention. Most of the existing approaches use supervised, trained on an annotated corpus. There are three main limitations with this type of approaches. First, an entity can be only classified with a class from a pre-defined list of classes used in the training phase. New entity classes, which might evolve over the time, are not considered. Second, the type might not be valid at the time of processing. While at one point of time an entity can be of type X, in another point of type it can be of type Y. And third, the entity type is usually not of a right granularity. In some cases it can be too general, while in other it can be too specific. In our work, we focused on exploiting public knowledge for entity classification in order to address these issues. In particular, we rely on Wikipedia as a source of information for the classification phase.

The entity classification is based on the Targeted Hypernym Discovery (THD) algorithm [94]. The goal of THD is to find a hypernym for a given entity; a word with a broader

\(^5\)Employed ESA implementation is available at: http://ticcky.github.io/esalib/
meaning. In our work, we adopt THD for the entity classification task where the extracted hypernym is considered as an entity type.

The hypernym discovery approach proposed here is based on hand-crafted lexico-syntactic patterns. The patterns, same as in the entity spotting phase, are defined as JAPE grammars, which are executed in the GATE framework. The JAPE grammars were defined based on an example set of 500 randomly chosen Wikipedia articles. The set of random articles was created using the “random” (list=random) feature of the Wikipedia Search API\(^6\). Further, we manually annotated the first hypernym in the first sentence of each article. We used this gold-standard set with annotated hypernyms to evaluate the quality of the hypernym extraction with the defined JAPE grammars. Then, in an iterative process, we improved the grammars, so that we achieve best performance. In our future work, we plan to work on an automated learning of these patterns.

In the past, lexico-syntactic patterns were primarily used on larger text corpora with the intent to discover all word-hypernym pairs in the collection. On this task, the state-of-the-art algorithm of Snow [95] achieves F-measure of 36%. However, with THD we apply lexico-syntactic patterns on a suitable document with the intent to extract one hypernym at a time. This approach is more successful – e.g. [96] report F1 measure of 0.851 with precision 0.969.

The entity classification is executed as follows. First, an entity mention is linked to a Wikipedia article. Next, the first sentence of the Wikipedia article is processed and the first hypernym is extracted. The hypernym extraction is performed using hand-crafted lexico-syntactic patterns defined as JAPE grammars. For example, for the Wikipedia article for Berlin (https://en.wikipedia.org/wiki/Berlin), the extracted hypernym will be “capital” since that is the first hypernym in the first sentence of the article: “Berlin is the capital and ...” is the word “capital”. Then, the extracted hypernym is linked to a DBpedia resource using the MFS approach. Finally, the extracted and linked hypernym can be considered as the type of the entity mentioned in the text.

### 4.1.3 Implementation

The system is implemented as a Web 2.0 application on top of the open source GATE framework\(^7\). The general system architecture is depicted in Figure 4.1. It consists of several components, each dedicated for a particular NER task.

The Entity Extraction module identifies entity candidates (noun phrases) in the input text. Based on the setting, the extraction process can be restricted to only extraction of named named entities (“Diego Maradona”), common entities (“football”), or both. As described in Section 4.1.2.1, the entity spotting is realized using lexico-syntactic patterns which are implemented as JAPE grammars and executed as part of a processing pipeline in the GATE framework. The entity spotting pipeline consists of a tokenizer, POS tagger and JAPE Transducer, which executes the pre-defined JAPE grammars. Depending on

---

\(^6\)API request for five random Wikipedia pages: https://en.wikipedia.org/w/api.php?action=query&list=random&rnlimit=5

\(^7\)https://gate.ac.uk/
4.1. Named Entity Recognition and Linking

![Figure 4.1: Architecture overview.](image)

The selected language, the pipeline is executing a POS tagger and a JAPE grammar for the given language. Currently, we define grammars for English, German and Dutch.

The **Entity Linking module** assigns a DBpedia resource URI describing the entity candidate. The module provides access for some methods described in Section 4.1.2.2. For the realization of the MFS based linking we maintain local Wikipedia mirror and specialized Lucene index for English, German and Dutch Wikipedias. We also provide support for the live Wikipedia endpoint which is used for real-time linking.

In addition, we also integrated a method called Surface Form Index (SFI) (see Section 4.1.5.4 for more information on the SFI method). It uses a custom entity candidate index which contains all surface forms found in the Wikipedia articles together with their candidates.

The **Entity Classification module** implements the entity classification approach described in Section 4.1.2.3. The classification is performed upon successful linking of the entity mention. The classification module is realized as a pipeline in the GATE framework which consists of a tokenizer, sentence splitter, POS tagger, noun phrase chunker and a JAPE transducer, which executes the pre-defined JAPE grammars. JAPE grammars have been developed for English, German and Dutch and are utilized based on the selected language of the input text. The POS tagger is also language dependent and executed based on the input text language. As described in Section 4.1.2.3, the classification is based on
4. Knowledge Extraction and Integration with Salient Linked Entities

the THD algorithm and it first extracts a hypernym from the Wikipedia article describing the entity and then the hypernym is linked with a DBpedia resources. Type mining can be performed on-the-fly either on-line from the live Wikipedia endpoint or from the endpoint of our maintained Wikipedia mirror. Currently, we maintain local Wikipedia mirrors for English, German and Dutch. We also integrate the Linked Hypernyms Dataset (LHD), which contains pre-computed types for DBpedia resources. In addition to the derived types, we also provide types from two prominent semantic knowledge bases, DBpedia and YAGO. The final set of types returned for an entity contains the mined types, as well as the types from DBpedia and YAGO. The system also enables users to choose the provenance of the types (THD, DBpedia, YAGO) and the knowledge base for mining types (local/live Wikipedia mirror).

The Entity Salience module is dedicated for identification of salient entities in text. Our system integrates a machine learning method for estimation of salience of each entity in the document. The method classifies each entity as most, less or not salient, and it assigns a confidence scores for the salience class. Section 4.2 describes in detail the developed method for entity salience.

The code for the whole system and the developed components is open-source under the GNU GPLv3 license.

4.1.4 The Web API

In order to expose the functionalities of the system, we developed a Web API. The Web API exposes the same functionalities as the user interface. The functionalities are controlled via a set of parameters. In particular, a user can specify: the language of the input text, provenance of types, knowledge base for mining types, preferred entity spotting and linking methods and the type of entities to extract (named and/or common entities). In addition, if required, a user can submit a list of recognized entities and classify them.

The Web API communicates the results in the NLP Interchange Format (NIF) and it can consume and produce results in the NIF format. In order to prevent abuse of the service, the API is secured with an API key. A user can obtain an API key by filling a request form. Currently, all API requests are subject to rate limits applied on the API key. We apply rate limits on a 24 hours basis (1 day) with users being able to submit up to 10,000 requests per day.

4.1.5 Evaluation and Validation

In this section, we present several experimental evaluations and a case study.

- An experimental evaluation which evaluates compares the performance of our NER system EntityClassifier.eu and three other state of the art systems on English con-
4.1. Named Entity Recognition and Linking

tent. The evaluation is performed on a manually created dataset, named "Czech
Traveler Dataset" (Section 4.1.5.1 [A.10]), which is based on textual annotations
from a travelogue-like photo collection.

- An experiment which evaluates the performance of our system on German and Dutch
texts. The evaluation is executed on a real-world data provided by the Netherlands
Institute for Sound and Vision (S&V) and the German national broadcaster Rund-
funk Berlin-Brandenburg (RBB) (Section 4.1.5.2) [A.14].

- Two experimental evaluations: at the TAC 2013 [A.8, A.13](see Section 4.1.5.3) and
TAC 2014 [A.7, A.14] (see Section 4.1.5.4) entity linking and discovery challenges.
The goal of these evaluations was to compare our NER system with other methods
which have been exclusively developed for these challenges.

- A case study where the focus is on extraction of concepts from video transcripts (see
Section 4.1.5.5 [A.11, A.15]). The case study was conducted on data from TRECVID,
a benchmarking activity for multimedia analysis.

4.1.5.1 Evaluation at the Czech Traveler Dataset

This section reports on a set of experiments which evaluate and compare the performance of
our Entityclassifier.eu NER system and three other state-of-the-art NER systems: DBpedia
Spotlight, Open Calais and Alchemy API. The contents of this section is based on a paper
[A.10] and contributions to a technical report [A.14].

Evaluation Setup

The evaluation has been executed on a manually created dataset named the "Czech Traveler
Dataset". The dataset is based on textual annotations from a travelogue-like photo
collection and it contains 101 named entities and 85 common entities. The experiments
were run for the Entityclassifier.eu system and three other systems: DBpedia Spotlight,
Open Calais and Alchemy API. We used the NERD tool (http://nerd.eurecom.fr) to
access these systems and process the content.

Next, the results were manually evaluated by two researchers with experience in text
mining. Following guidelines have been provided. Entity spotting: named entity or com-
mon entity was considered as correctly extracted only if there was a full textual match;
extracting "Maradona" instead of "Diego Maradona" would be considered as an error. Entity
classification: the distance of the type was not considered as long as it was cor-
rect; manager is as good a type for Maradona as person. In other words, both types,
person and manager, were considered as correct types for the entity Maradona. Entity
linking: the target of the link must describe the entity given context; the documentary
"dbpedia:Maradona_by_Kusturica" would be considered as incorrect replacement for "db-
pedia:Diego_Maradona". We evaluated the entity spotting, classification and linking of
named entities and common entities.

11https://ner.vse.cz/datasets/czechtraveler/
4. Knowledge Extraction and Integration with Salient Linked Entities

Results

Below we report on the results from the evaluation and report on the precision, recall and F1 score for each system and task.

Table 4.1 shows the results from the evaluation of spotting of named entities for Entity-classifier.eu and the other three systems. The results show that our unsupervised approach for entity spotting based on lexico-syntactic patterns outperformed the other three systems. Our system achieved 0.66 F1 score, AlchemyAPI achieved second best results with 0.57 F1, followed by OpenCalais with 0.42 F1 and DBpedia Spotlight with 0.34 F1.

Table 4.1: Evaluation results for spotting of named entities.

<table>
<thead>
<tr>
<th>System</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>EntityClassifier.eu</td>
<td>0.73</td>
<td>0.59</td>
<td>0.66</td>
</tr>
<tr>
<td>DBpedia Spotlight</td>
<td>0.47</td>
<td>0.27</td>
<td>0.34</td>
</tr>
<tr>
<td>OpenCalais</td>
<td>0.62</td>
<td>0.32</td>
<td>0.42</td>
</tr>
<tr>
<td>AlchemyAPI</td>
<td>0.75</td>
<td>0.46</td>
<td>0.57</td>
</tr>
</tbody>
</table>

In Table 4.2 we summarize the results from classification of named entities. According to the results, our unsupervised approach for entity classification based on real-time mining of types achieved best results. Our system achieved 0.66 F1, followed by OpenCalais and AlchemyAPI with 0.45 F1 and DBpedia Spotlight with 0.41 F1.

Table 4.2: Evaluation results for classification of named entities.

<table>
<thead>
<tr>
<th>System</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>EntityClassifier.eu</td>
<td>0.77</td>
<td>0.57</td>
<td>0.66</td>
</tr>
<tr>
<td>DBpedia Spotlight</td>
<td>1.00</td>
<td>0.26</td>
<td>0.41</td>
</tr>
<tr>
<td>OpenCalais</td>
<td>0.97</td>
<td>0.30</td>
<td>0.45</td>
</tr>
<tr>
<td>AlchemyAPI</td>
<td>0.84</td>
<td>0.31</td>
<td>0.45</td>
</tr>
</tbody>
</table>

Table 4.3 shows the results from the evaluation of entity linking only for Entity-classifier.eu and DBpedia Spotlight, since at the time of execution of the experiments, OpenCalais and AlchemyAPI did not provide entity links. According to the results, our system outperformed DBpedia Spotlight. EntityClassifier.eu achieved 0.67 F1 score, compared to 0.43 F1 by DBpedia Spotlight.

Table 4.3: Evaluation results for linking of named entities.

<table>
<thead>
<tr>
<th>System</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>EntityClassifier.eu</td>
<td>0.80</td>
<td>0.59</td>
<td>0.67</td>
</tr>
<tr>
<td>DBpedia Spotlight</td>
<td>0.97</td>
<td>0.28</td>
<td>0.43</td>
</tr>
</tbody>
</table>
We have also evaluated the performance of our and the three other systems for spotting, classification and linking of common entities. Table 4.4 summarizes the results from the evaluation. According to the results, our system has shown same performance as DBpedia Spotlight (0.60 F1), but higher precision (0.71 vs. 0.63) and lower recall (0.52 vs. 0.58). OpenCalais and AlchemyAPI have shown very poor performance with 0.14 and 0.10 F1, respectively. The results show that OpenCalais and AlchemyAPI are not optimized for spotting of common entities.

<table>
<thead>
<tr>
<th>System</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>EntityClassifier.eu</td>
<td>0.71</td>
<td>0.52</td>
<td>0.60</td>
</tr>
<tr>
<td>DBpedia Spotlight</td>
<td>0.63</td>
<td>0.58</td>
<td>0.60</td>
</tr>
<tr>
<td>OpenCalais</td>
<td>0.50</td>
<td>0.08</td>
<td>0.14</td>
</tr>
<tr>
<td>AlchemyAPI</td>
<td>0.42</td>
<td>0.06</td>
<td>0.10</td>
</tr>
</tbody>
</table>

In Table 4.5 we summarize the results from the evaluation of classification of common entities. The best results were achieved by DBpedia Spotlight with 0.57 F1, followed by our system EntityClassifier.eu with 0.51, OpenCalais with 0.17 and AlchemyAPI with 0.11.

<table>
<thead>
<tr>
<th>System</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>EntityClassifier.eu</td>
<td>0.63</td>
<td>0.44</td>
<td>0.51</td>
</tr>
<tr>
<td>DBpedia Spotlight</td>
<td>0.92</td>
<td>0.41</td>
<td>0.57</td>
</tr>
<tr>
<td>OpenCalais</td>
<td>0.89</td>
<td>0.09</td>
<td>0.17</td>
</tr>
<tr>
<td>AlchemyAPI</td>
<td>0.50</td>
<td>0.06</td>
<td>0.11</td>
</tr>
</tbody>
</table>

Finally, Table 4.6 reports on the performance of EntityClassifier.eu and DBpedia Spotlight for the task of linking common entities. We do not report on the performance of OpenCalais and AlchemyAPI since at the time of execution of the experiments they did not provide entity links. In this experiment, DBpedia Spotlight achieved better results compared to EntityClassifier.eu (0.69 F1 vs. 0.61 F1).

<table>
<thead>
<tr>
<th>System</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>EntityClassifier.eu</td>
<td>0.75</td>
<td>0.52</td>
<td>0.61</td>
</tr>
<tr>
<td>DBpedia Spotlight</td>
<td>0.92</td>
<td>0.55</td>
<td>0.69</td>
</tr>
</tbody>
</table>

In overall, our system produced consistently better results in all tasks than the other three systems except for linking and classification of common entities where DBpedia Spotlight has shown better results.
4. Knowledge Extraction and Integration with Salient Linked Entities

4.1.5.2 Evaluation and validation on a LinkedTV Scenario

This evaluation is based on contributions to a technical report [A.14] and it describes several experimental evaluations on the performance of our NER system Entityclassifier.eu\(^\text{12}\) (Section 4.1.3). The evaluation was conducted on real-world datasets provided from two organizations participating in the LinkedTV EU project. The goal of this evaluation was to assess the quality of several NER and EL methods implemented in our Entityclassifier.eu NER system on different languages than English, i.e German and Dutch. For the performance of the NER system on English content please see Sections 4.1.5.1, 4.1.5.3 and 4.1.5.4. The evaluation was performed on subtitles associated with the LinkedNews and LinkedCulture videos, which were provided and annotated by the Netherlands Institute for Sound and Vision (S&V) and the German national broadcaster Rundfunk Berlin-Brandenburg (RBB). From the S&V dataset we processed the subtitles for the Horse Painting scene, which is part of the Museum Martena episode. From the RBB dataset we processed subtitles from an episode about asparagus. In the following, we describe the annotation process, the methods evaluated for entity spotting and linking, and present the results.

Ground-truth Datasets

The subtitles for processing were selected and annotated by professionals from Sound&Vision and RBB. The professionals were asked to select sequences of videos lasting around 15 min in total. Next, we generated an annotation sheet containing the subtitles content. From the provided subtitles, we excluded words which indicate a specific sound or strange happening (e.g., laughing or audience gasps). The annotators were given strict annotation guidelines and their task was to:

- identify each entity occurrence in the given subtitle by providing its surface form (i.e. the text used to refer to an entity in text),
- provide URL of Wikipedia page which describes the entity, and
- provide information whether the entity refers to a named entity (proper nouns) or common entity (nouns with a preceding modifier).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Num. of entities</th>
<th>Named entities</th>
<th>Common entities</th>
<th>Video length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sound &amp; Vision</td>
<td>289</td>
<td>42</td>
<td>247</td>
<td>14 min 55 sec</td>
</tr>
<tr>
<td>RBB</td>
<td>397</td>
<td>75</td>
<td>322</td>
<td>13 min 50 sec</td>
</tr>
</tbody>
</table>

Evaluated Approaches

\(^\text{12}\)Available at http://entityclassifier.eu/
4.1. Named Entity Recognition and Linking

For entity spotting, we evaluated our unsupervised entity spotting approached based on lexico-syntactic patterns (Section 4.1.2.1). Will will further refer to this approach as GRAM. Further, we evaluated following entity linking approaches:

- **Basic Lucene index** (LB) - linking based on specialized Lucene index, which extends the Apache Lucene search API (Section 4.1.2.2). This approach primarily ranks pages based on the number of backlinks and the Wikipedia articles' titles.

- **Lucene Skip Disambiguation** (LSD) - same as the previous, only as a correct link it considers the first non-disambiguation page.

- **Surface Form Index** (SFI) - this approach uses a custom entity candidate index. The candidate index contains all surface forms found in Wikipedia articles together with their candidates. This approach has been developed by Ivo Lasek and it has been considered as baseline.

- **Surface Form Similarity** (SFS) - this approach first performs entity linking with the SFI and LSD. And then, the article with the most similar title to the entity surface form is considered as correct. For measuring similarity we opted for the widely used Jaro-Winkler string similarity measure.

Results

The evaluation was performed using the GERBIL [97] benchmarking framework for NER systems. For the evaluation we run the A2KB experiment [97], which is an experiment which evaluates the performance of entity spotting and linking. We report on three metrics computed by the GERBIL framework: micro precision, micro recall and micro F-measure. The micro measures provide information on the performance over the set of all annotations inside the dataset, while the macro measures evaluate average performance per document. More details on the metrics can be found in the description paper of the GERBIL framework [97]. The macro measures are not reported, since scores obtained on such short documents (one document corresponds to one subtitle fragment) are not very meaningful, as many of the documents have no ground-truth annotations, which results in increased macro score measures.

Table 4.8 shows the results from the evaluation on the Dutch dataset with focus on the named entities only. The results show that the best F1 micro score 0.4242 was achieved by the approach which uses the Surface Form Index (SFI) for entity linking.

Table 4.8: Evaluation results for Dutch. An experiment type A2KB - named entities only.

<table>
<thead>
<tr>
<th>spotting/linking</th>
<th>Micro F1</th>
<th>Micro P</th>
<th>Micro R</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRAM/LB</td>
<td>0.3333</td>
<td>0.2400</td>
<td>0.5455</td>
</tr>
<tr>
<td>GRAM/LSD</td>
<td>0.3333</td>
<td>0.2400</td>
<td>0.5455</td>
</tr>
<tr>
<td>GRAM/SFS</td>
<td>0.3889</td>
<td>0.2800</td>
<td>0.6364</td>
</tr>
<tr>
<td>GRAM/SFI</td>
<td><strong>0.4242</strong></td>
<td>0.3182</td>
<td>0.6364</td>
</tr>
</tbody>
</table>
4. Knowledge Extraction and Integration with Salient Linked Entities

Table 4.9 shows the results from the evaluation on the Dutch dataset which contains both, named and common entities. In this evaluation, the best micro 0.1958 F1 score was achieved by the Surface Form Similarity (SFS) based approach, followed by the Basic Lucene (LB) based approach with 0.1824 F1.

Table 4.9: Evaluation results for Dutch. An experiment type A2KB - named and common entities.

<table>
<thead>
<tr>
<th>spotting/linking</th>
<th>Micro F1</th>
<th>Micro P</th>
<th>Micro R</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRAM/LB</td>
<td>0.1824</td>
<td>0.1079</td>
<td>0.5882</td>
</tr>
<tr>
<td>GRAM/LSD</td>
<td>0.1763</td>
<td>0.1043</td>
<td>0.5686</td>
</tr>
<tr>
<td>GRAM/SFS</td>
<td><strong>0.1958</strong></td>
<td>0.1154</td>
<td>0.6471</td>
</tr>
</tbody>
</table>

Table 4.10 shows the results from the evaluation on the Dutch dataset which contains common entities only. The results show that the best micro F1 score 0.1358 was achieved by the Surface Form Similarity (SFS) approach.

Table 4.10: Evaluation results for Dutch. An experiment type A2KB - common entities only.

<table>
<thead>
<tr>
<th>spotting/linking</th>
<th>Micro F1</th>
<th>Micro P</th>
<th>Micro R</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRAM/LB</td>
<td>0.1323</td>
<td>0.0746</td>
<td>0.5862</td>
</tr>
<tr>
<td>GRAM/LSD</td>
<td>0.1245</td>
<td>0.0702</td>
<td>0.5517</td>
</tr>
<tr>
<td>GRAM/SFS</td>
<td><strong>0.1358</strong></td>
<td>0.0763</td>
<td>0.6207</td>
</tr>
</tbody>
</table>

Table 4.11 shows the results from the evaluation on the German dataset with focus on the named entities only. According to the results the methods which use the Lucene index for linking achieved the best micro 0.5349 F1.

Table 4.11: Evaluation results for German. An experiment type A2KB - named entities only.

<table>
<thead>
<tr>
<th>spotting/linking</th>
<th>Micro F1</th>
<th>Micro P</th>
<th>Micro R</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRAM/LB</td>
<td><strong>0.5349</strong></td>
<td>0.5897</td>
<td>0.4894</td>
</tr>
<tr>
<td>GRAM/LSD</td>
<td><strong>0.5349</strong></td>
<td>0.5897</td>
<td>0.4894</td>
</tr>
<tr>
<td>GRAM/SFS</td>
<td>0.5176</td>
<td>0.5789</td>
<td>0.4681</td>
</tr>
<tr>
<td>GRAM/SFI</td>
<td>0.5301</td>
<td>0.6111</td>
<td>0.4681</td>
</tr>
</tbody>
</table>

Table 4.12 shows the results from the evaluation on the German dataset which contains both, named and common entities. It can be observed that the best micro score 0.4207 was achieved by the method which performs entity linking based on the Surface Form Similarity (SFS) method.

Table 4.12: Evaluation results for German. An experiment type A2KB - named and common entities.

<table>
<thead>
<tr>
<th>spotting/linking</th>
<th>Micro F1</th>
<th>Micro P</th>
<th>Micro R</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRAM/LB</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRAM/LSD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRAM/SFS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GRAM/SFI</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 4.12: Evaluation results for German. An experiment type A2KB - named and common entities.

<table>
<thead>
<tr>
<th>spotting/linking</th>
<th>Micro F1</th>
<th>Micro P</th>
<th>Micro R</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRAM/LB</td>
<td>0.3680</td>
<td>0.2585</td>
<td>0.6387</td>
</tr>
<tr>
<td>GRAM/LSD</td>
<td>0.3583</td>
<td>0.2500</td>
<td>0.6323</td>
</tr>
<tr>
<td>GRAM/SFS</td>
<td>0.4207</td>
<td>0.2946</td>
<td>0.7355</td>
</tr>
</tbody>
</table>

Table 4.13 shows the results from the evaluation on the German dataset which contains common entities only. In this experiment, the best micro 0.3545 has been achieved by the Surface Form Similarity (SFS) linking method.

Table 4.13: Evaluation results for German. An experiment type A2KB - common entities only.

<table>
<thead>
<tr>
<th>spotting/linking</th>
<th>Micro F1</th>
<th>Micro P</th>
<th>Micro R</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRAM/LB</td>
<td>0.2993</td>
<td>0.1955</td>
<td>0.6389</td>
</tr>
<tr>
<td>GRAM/LSD</td>
<td>0.2820</td>
<td>0.1841</td>
<td>0.6019</td>
</tr>
<tr>
<td>GRAM/SFS</td>
<td>0.3545</td>
<td>0.2321</td>
<td>0.7500</td>
</tr>
</tbody>
</table>

Summary

In the “named entity only” evaluation, our best performing approach achieved 0.5349 micro F1 score for German, and 0.4242 micro F1 score for Dutch. Worse results, 0.3545 F1 for German and 0.1358 F1 for Dutch, were obtained on the more ambiguous common entities. For both entity types, named and common entities, the best micro F1 scores are at 0.4207 for German and at 0.1958 for Dutch. This lower overall result is caused by the prevalence of common entities in the corpus, which are in general more ambiguous.

According to the results from the experiments, our approaches work better for German than Dutch. One of the reasons for the better performance on German than Dutch texts, could be that our pattern based spotting and most-frequent-sense based linking approaches are better suited for the German language than Dutch. In other words, using lexico-syntactic patterns it is easier to capture named entity mentions for German than Dutch. In our future work, we will work on improvement of our lexico-syntactic patterns used for spotting NER by adopting them on similar content as the one used for the evaluation.

4.1.5.3 Evaluation at TAC 2013

In Section 4.1.5.1, we have executed an evaluation and compared the performance of our NER system Entityclassifier.eu on English to three other state of the art NER systems: DBpedia Spotlight, Open Calais and AlchemyAPI. In Section 4.1.5.2 we have executed another experiment and evaluated our system on German and Dutch texts. In this section,
we report on the evaluation of Entityclassifier.eu (Section 4.1) on the TAC KBP 2013 Entity Linking task\textsuperscript{13}. The purpose of this evaluation was to evaluate our NER system on another dataset, on different focus queries and compare it to additional methods which have been exclusively developed for this challenge. This section is based on a paper [A.8] and contributions to a technical report [A.13].

At the challenge we evaluated various modifications and combinations of the Most-Frequent-Sense (MFS) based linking and the Explicit Semantic Analysis (ESA) based linking, which is supported by the ESA representation model [91]. For the challenge we submitted 9 submissions in total, from which 5 used the textual context of the entities, and 4 submissions did not. Surprisingly, the MFS method based on the Wikipedia Search has proved to be the most effective approach – it achieved the best 0.555 $B_{3+}^2$ F1 score from all our submissions and it achieved highest 0.677 $B_{3+}^2$ F1 score for Geo-Political (GPE) entities. In addition, the ESA based method achieved best 0.483 $B_{3+}^2$ F1 for Organization (ORG) entities.

We first describe how the provided TAC dataset was prepared and linked with our Wikipedia (resp. DBpedia) knowledge base. Next, we list the evaluated entity linking methods and provide description for each of the 9 submissions. Finally, we present and discuss the results from the evaluation.

The Entity Linking Task Description

The entity linking task, as described by the challenge organizers\textsuperscript{14}, is defined as task of linking entity mentions in a document corpora to entities in a reference KB. If the entity is not already in the reference KB, a new entity node should be added to the KB. Each participation team was given a set of 2190 queries consisting of a queryID, docID, name (name mention of an entity) and beginning and end entity offsets in the document.

Further, the system performing the entity linking task had to output the results providing information about the queryID, KB_link (or NIL entity identifier, if the entity was not present in the KB) and a confidence value.

Data Preparation

Since the TAC KBP reference knowledge base uses custom identifiers of the entities (e.g. E0522900) and our systems identify the entities with DBpedia URIs, it was necessary to map these identifiers.

In the TAC KBP knowledge base, each entity entry provides information about the custom identifier of the entity, and the path URL segment of a Wikipedia article describing the entity (e.g., entity with URL Sam_Butler and id E0522900). Since DBpedia also derives its URIs from the URIs of the Wikipedia articles, we used the URLs of the Wikipedia articles describing the entities to map them to DBpedia. For example, the entity in the TAC KBP KB with identifier E0522900 was mapped to DBpedia URI

\textsuperscript{13}https://tac.nist.gov//2013/KBP/EntityLinking/index.html
\textsuperscript{14}http://www.nist.gov/tac/2013/KBP/EntityLinking/
4.1. Named Entity Recognition and Linking

http://dbpedia.org/resource/Sam_Butler. This way we could relate the DBpedia URI identifiers of our systems with the entity identifiers in the TAC reference knowledge base.

**Evaluated Approaches**

We have evaluated three variants of a most-frequent sense (MFS) approach and one novel method based on the ESA representation model which has been exclusively developed for the entity linking challenge.

All methods follow the three-steps approach defined as follows. First, it applies *candidate selection*, where a set of entity candidates are retrieved for the given entity. Second, it performs the *disambiguation*, where one candidate from the candidates list is selected as the correct one. Finally, selected entity candidate is linked, i.e. a reference in the TAC KBP knowledge base is identified. If the entity is not found in the KB, then, a new entity NIL node is added. Below we describe each used method for entity linking followed by a description of each submission. The list of entity linking methods evaluated at the challenge is as follows:

- **Most Frequent Sense (MFS) based linking** - links an entity with the most-frequent-sense entity found in the reference knowledge base. See Section 4.1.2.2 for more information on the MFS method. We have implemented three slightly different variants of the MFS method. See submission #1-#3 for the description of the three variants.

- **Explicit Semantic Analysis (ESA) based linking** - a context-based linking where the contextual similarity of the for the entity mention and each entity candidate is estimated using the ESA representation model. See Section 4.1.2.2 for more information on the ESA method.

In addition to the entity linking methods listed above, we have also evaluated the performance of the *Entity Co-occurrence (ECC) method* which has been considered as baseline to compare with. ECC is a context based method and it relies on the assumption that entities often co-occur in texts. *Note: the ECC method has been developed by Ivo Lašek.* More details on the method can be found in [A.8].

**Submissions Description**

For the TAC KBP 2013 Entity Linking task we have submitted 9 runs based on different variations of the MFS, ECC and ESA methods. Below we provide detailed description for each individual submission. The first four runs rely on the MFS linking approach, the fifth run relies on the entity ECC linking approach, the sixth on the ESA linking approach, the seventh is a merged submission of the ECC and ESA linking approaches, the eight is a combination of the MFS and ESA linking approaches, and finally, the ninth is a merged submission of the MFS, ECC and ESA linking approaches.

**Run #1.** This run relies on the MFS approach to perform the linking. In this run, each entity mention was considered as an entity, so the entity spotting step was not performed.
4. Knowledge Extraction and Integration with Salient Linked Entities

To realize the MFS approach we used the English Wikipedia Search API. For the entity candidate selection step we run a Wikipedia search for the most five relevant articles describing the entity in question. The article with the highest rank in the result list was considered as the correct entity. Finally, this entity was linked with an entity in the knowledge base. If it was not found, then, a new entity NIL node was created. This run did access the Web during the processing and it did not use the entity context text (wiki_text element text) neither it uses the entity offsets.

Run #2. This run also relies on the MFS linking approach. Compared to the previous run, in this run we used the Entityclassifier.eu\textsuperscript{15} NER system to perform the linking. In this run, each entity mention was submitted to the NER system. The system decides whether the string represents an entity. In a positive scenario the system disambiguates the entity by running a Wikipedia search on the API for the local English Wikipedia mirror. Similarly like in the previous run, the highest ranked result is considered as the correct entity. The NER system finally returns a DBpedia resource URI which describes the entity. This entity URI is then linked to the entity in the reference knowledge base. If it was not found, then, a new entity NIL node was created. Note that this run did not access the Web during the processing and it does not use the entity context text (wiki_text element text), neither it uses the entity offsets.

Run #3. This run, same as the previous two runs, relies on the MFS linking approach. This run uses a local Lucene index created for an English Wikipedia snapshot, as of 18/9/2012. Note that this Lucene index is also used by the Wikipedia Search API. Each entity mention was searched in the index and the first returned result was considered as the correct entity. Unlike the previous runs, this run provides direct reproducibility, as it involves no third-party hosted software or data. Since this run uses a local Lucene index, it did not access the Web during the processing, neither it uses the entity context text or the entity offsets.

Run #4. This is a merged submission of the previous three MFS linking approaches. We experienced that merging the results produced conflicts very often. These conflicts were resolved based on the performance of the individual methods on the TAC 2012 corpus. Thus, run #2 had the highest priority followed by the run #1, and finally, run #3. Since this run merges results from two submissions that do not access the Web and one that accesses the Web, it can be considered that this run accesses the Web too. On the other hand, this submission does not use the entity context text.

Run #5. This run relies on the ECC entity linking method and we used the SemiTags NER system to perform the linking. This run is considered as baseline. More details on the method can be found in [A.8].

Run #6. This run relies on the ESA entity linking method. In this method, each entity mention is considered as an entity. For each query, top five results returned by Wikipedia Search API are used as entity candidates. Next, the first paragraph of each of

\textsuperscript{15}Note that for the run #2 we used a local instance of the NER system so we consider this run as run that does not access the Web. The API documentation of the Entityclassifier.eu NER system is publicly available at: http://entityclassifier.eu/thd/docs/
these candidate articles is retrieved. Finally, the ESA\textsuperscript{16} method is used to compute the similarity of the first paragraph of each entity candidate with the entity context text. The entity context text is a 800 characters long text constructed from the wiki\textunderscore text element. Same as in the previous submission, it consists the entity and 400 characters preceding and following the entity. After computing the similarity between each first entity paragraph and the entity context text, the entity (first Wikipedia article paragraph) with the highest similarity score is considered as correct and it was linked with the entity in the reference KB. This run does not access the Web during the evaluation.

**Run #7.** This is a merged submission of the ESA and the ECC linking methods which uses the wiki\textunderscore text. A higher priority was given to the ECC method when resolving the conflicts. Since none of the two submissions access the Web, it can be considered that this run does not access the Web either.

**Run #8.** This run combines the MFS and the ESA entity linking approaches. In this run, for each query, we used the Wikipedia Search API to retrieve a list of entity candidates. We used only the first result as a candidate (cf. run #1). Next, the ESA method was used to compute the similarity of the first paragraph of the Wikipedia article describing the entity candidate with the entity context text. As the context of the entity 400 characters preceding and following the query (800 altogether) was used. In addition, threshold of 0.15 was set for the ESA linking method. If the similarity score was higher or equal to 0.15, then the entity was considered as correct and the entity was linked with the entity in the reference KB. Otherwise, it was considered as incorrect and a new entity NIL node was added. This run accesses the Web during the evaluation and it is using the entity context text (wiki\textunderscore text element).

**Run #9.** This run is a merged submission of four individual submissions. The conflicts were resolved by assigning priority to each individual submission. The highest priority was given to the run #2 (MFS with Entityclassifier.eu NER), followed by run #5 (ECC), run #6 (ESA) and run #1 (MFS baseline). Since some runs access the Web and also use the entity context (wiki\textunderscore text), it can be considered that this submission accesses the Web and uses the entity context too.

In all the runs the basic "exact name" NIL clustering technique was only used.

**Evaluation Metrics**

In the TAC 2013 KBP Entity Linking task the systems were evaluated using three scoring metrics. The micro-average ($\mu$AVG) [98], the B-cubed cluster scoring ($B^3$) [99] and the B-cubed+ modification ($B^{3+}$) [100]. The main difference between the B-cubed and B-cubed+ scoring metrics is that for non-NIL queries, B-cubed+ not only considers the quality of clustering as in B-cubed, but also measures the accuracy of linking these queries to correct KB entries.

**Results**

---

\textsuperscript{16}Employed ESA implementation is available at: http://ticcky.github.io/esalib/
4. Knowledge Extraction and Integration with Salient Linked Entities

We report all three scoring metrics for each of our submissions, as well we report on the performance of our methods for the focus queries, i.e. how well our methods perform for entities of type person, organization, etc. Note that we also report on the precision and recall for each metric and the median value for all participating teams.

In Table 4.14 we provide the overall performance achieved of each individual run. The results show that in overall, the MFS linking method (cf. run #1) performed the best, achieving 0.707 B-cubed F1 score, 0.555 B-cubed+ F1 score and highest B-cubed+ precision score 0.653. The highest B-cubed precision score was achieved by the ECC linking method with score of 0.912.

<table>
<thead>
<tr>
<th>Id</th>
<th>μAVG</th>
<th>B³ P</th>
<th>B³ R</th>
<th>B³ F1</th>
<th>B³⁺ P</th>
<th>B³⁺ R</th>
<th>B³⁺ F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>run #1</td>
<td>0.737</td>
<td>0.870</td>
<td>0.596</td>
<td>0.707</td>
<td>0.653</td>
<td>0.483</td>
<td>0.555</td>
</tr>
<tr>
<td>run #2</td>
<td>0.686</td>
<td>0.821</td>
<td>0.515</td>
<td>0.633</td>
<td>0.568</td>
<td>0.387</td>
<td>0.461</td>
</tr>
<tr>
<td>run #3</td>
<td>0.727</td>
<td>0.844</td>
<td>0.593</td>
<td>0.696</td>
<td>0.632</td>
<td>0.471</td>
<td>0.540</td>
</tr>
<tr>
<td>run #4</td>
<td>0.733</td>
<td>0.835</td>
<td>0.570</td>
<td>0.678</td>
<td>0.622</td>
<td>0.461</td>
<td>0.530</td>
</tr>
<tr>
<td>run #5</td>
<td>0.611</td>
<td>0.912</td>
<td>0.428</td>
<td>0.582</td>
<td>0.558</td>
<td>0.292</td>
<td>0.383</td>
</tr>
<tr>
<td>run #6</td>
<td>0.625</td>
<td>0.896</td>
<td>0.500</td>
<td>0.642</td>
<td>0.562</td>
<td>0.358</td>
<td>0.437</td>
</tr>
<tr>
<td>run #7</td>
<td>0.658</td>
<td>0.901</td>
<td>0.499</td>
<td>0.642</td>
<td>0.604</td>
<td>0.373</td>
<td>0.462</td>
</tr>
<tr>
<td>run #8</td>
<td>0.717</td>
<td>0.887</td>
<td>0.546</td>
<td>0.676</td>
<td>0.640</td>
<td>0.433</td>
<td>0.517</td>
</tr>
<tr>
<td>run #9</td>
<td>0.704</td>
<td>0.850</td>
<td>0.580</td>
<td>0.690</td>
<td>0.610</td>
<td>0.461</td>
<td>0.525</td>
</tr>
</tbody>
</table>

Compared with the reported median B-cubed+ F1 score at the challenge, the MFS based linking submissions (#1 - #4) achieved similar score for all the queries (see Table 4.15). However, for the focus queries targeting GPE entities the MFS submission achieved significantly better B-cubed+ F1 score 0.677 compared with the reported median at the challenge 0.552. A better B-cubed+ F1 compared to the median was also achieved for the focus queries targeting entities in discussion fora (0.539 compared to median 0.488).

Surprisingly, the MFS method based submissions, which do not use the entity context (wiki_element text) achieved better results compared with the ECC and ESA method based submissions, which use the entity context text.

At the challenge participated 27 teams, and in Table 4.16 we provide the results achieved by the best performing run of each team. The results show that our best performing run at the challenge achieved 0.555 B³⁺ F1 score, while the best performing run 0.764 B³⁺ F1 and the worst performing run 0.203 B³⁺ F1 score. It can be also observed that our most frequent sense based approach outperformed two other supervised approaches, which rely on the support vector machine (0.493 B³⁺ F1, team 18) and conditional random fields
Table 4.15: Comparison of the highest and median $B^3+$ F1 scores achieved at the challenge with the highest score achieved by our submissions.

<table>
<thead>
<tr>
<th>Focus Queries</th>
<th>Highest $B^3+$ F1</th>
<th>Median $B^3+$ F1</th>
<th>Our highest $B^3+$ F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>0.746</td>
<td>0.574</td>
<td>0.555</td>
</tr>
<tr>
<td>in KB</td>
<td>0.722</td>
<td>0.554</td>
<td>0.595</td>
</tr>
<tr>
<td>not in KB</td>
<td>0.777</td>
<td>0.566</td>
<td>0.601</td>
</tr>
<tr>
<td>NW - Newswire docs</td>
<td>0.829</td>
<td>0.645</td>
<td>0.586</td>
</tr>
<tr>
<td>WB - Web docs</td>
<td>0.678</td>
<td>0.525</td>
<td>0.484</td>
</tr>
<tr>
<td>DF - Discussion Fora docs</td>
<td>0.662</td>
<td>0.488</td>
<td>0.539</td>
</tr>
<tr>
<td>PER</td>
<td>0.778</td>
<td>0.627</td>
<td>0.501</td>
</tr>
<tr>
<td>ORG</td>
<td>0.737</td>
<td>0.542</td>
<td>0.483</td>
</tr>
<tr>
<td>GPE</td>
<td>0.746</td>
<td>0.552</td>
<td>0.677</td>
</tr>
</tbody>
</table>

learning (0.478 $B^3+$ F1, team 19) model. The results also show that our most frequent based approach shows better results than two unsupervised approaches; an approach considering only very minimal information such as the entity authority file and a plain text from the targeted domain (0.412 $B^3+$ F1, team 25), and an another unsupervised clustering based approach (0.310 $B^3+$ F1, team 26). Overall, our best performing run (0.555 $B^3+$ F1) achieved score close to the challenge median score 0.574 $B^3+$ F1 (see Table 4.15) and better score than the challenge average score 0.475 $B^3+$ F1 (arithmetic mean). Note that our main goal with our participation in the challenge was to evaluate the performance of the Entityclassifier.eu system on datasets from different domains and for different focus queries and compare to other approaches, which have been exclusively developed for the TAC challenge.

The queries in the Entity Linking task were targeting entities of three different types. Entities of type Person (PER), Organization (ORG) and Geo-Political entities (GPE). For the task, each participating team received 2,190 queries in total where 686 queries were targeting entities of type person, 701 targeting organizations and 803 targeting geo-political entities. Tables 4.17, 4.18 and 4.19 summarize the achieved results related to these focus queries for each individual submission.
4. Knowledge Extraction and Integration with Salient Linked Entities

Table 4.16: Results achieved by each team with their best performing run. Brief description is provided only for the approaches which are accompanied with a TAC 2013 paper.

<table>
<thead>
<tr>
<th>System</th>
<th>B³⁺ P</th>
<th>B³⁺ R</th>
<th>B³⁺ F1</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team 1</td>
<td>0.813</td>
<td>0.689</td>
<td>0.746</td>
<td>machine learning method</td>
</tr>
<tr>
<td>Team 2</td>
<td>0.806</td>
<td>0.662</td>
<td>0.727</td>
<td>supervised entity linking</td>
</tr>
<tr>
<td>Team 3</td>
<td>0.763</td>
<td>0.668</td>
<td>0.712</td>
<td>n/a</td>
</tr>
<tr>
<td>Team 4</td>
<td>0.776</td>
<td>0.628</td>
<td>0.694</td>
<td>supervised wikifier</td>
</tr>
<tr>
<td>Team 5</td>
<td>0.792</td>
<td>0.601</td>
<td>0.684</td>
<td>markov logic networks</td>
</tr>
<tr>
<td>Team 6</td>
<td>0.785</td>
<td>0.584</td>
<td>0.670</td>
<td>probabilistic information retrieval</td>
</tr>
<tr>
<td>Team 7</td>
<td>0.771</td>
<td>0.550</td>
<td>0.642</td>
<td>supervised, random forest</td>
</tr>
<tr>
<td>Team 8</td>
<td>0.746</td>
<td>0.555</td>
<td>0.637</td>
<td>combination; supervised and unsupervised</td>
</tr>
<tr>
<td>Team 9</td>
<td>0.743</td>
<td>0.555</td>
<td>0.635</td>
<td>clustering based coreference chains</td>
</tr>
<tr>
<td>Team 10</td>
<td>0.826</td>
<td>0.508</td>
<td>0.629</td>
<td>n/a</td>
</tr>
<tr>
<td>Team 11</td>
<td>0.744</td>
<td>0.518</td>
<td>0.611</td>
<td>n/a</td>
</tr>
<tr>
<td>Team 12</td>
<td>0.736</td>
<td>0.506</td>
<td>0.600</td>
<td>supervised, multi-NER based</td>
</tr>
<tr>
<td>Team 13</td>
<td>0.719</td>
<td>0.509</td>
<td>0.596</td>
<td>unsupervised; semantic relatedness</td>
</tr>
<tr>
<td>Team 14</td>
<td>0.718</td>
<td>0.498</td>
<td>0.588</td>
<td>logistic regression classification</td>
</tr>
<tr>
<td>Team 15</td>
<td>0.686</td>
<td>0.485</td>
<td>0.568</td>
<td>supervised learning</td>
</tr>
<tr>
<td><strong>Our best run</strong></td>
<td>0.653</td>
<td>0.483</td>
<td>0.555</td>
<td>Most-frequent-sense based</td>
</tr>
<tr>
<td>Team 17</td>
<td>0.640</td>
<td>0.405</td>
<td>0.496</td>
<td>n/a</td>
</tr>
<tr>
<td>Team 18</td>
<td>0.649</td>
<td>0.397</td>
<td>0.493</td>
<td>supervised; SVM</td>
</tr>
<tr>
<td>Team 19</td>
<td>0.597</td>
<td>0.399</td>
<td>0.478</td>
<td>supervised: Conditional Random Fields</td>
</tr>
<tr>
<td>Team 20</td>
<td>0.619</td>
<td>0.385</td>
<td>0.475</td>
<td>n/a</td>
</tr>
<tr>
<td>Team 21</td>
<td>0.561</td>
<td>0.407</td>
<td>0.472</td>
<td>n/a</td>
</tr>
<tr>
<td>Team 22</td>
<td>0.587</td>
<td>0.392</td>
<td>0.470</td>
<td>n/a</td>
</tr>
<tr>
<td>Team 23</td>
<td>0.572</td>
<td>0.366</td>
<td>0.446</td>
<td>supervised learning</td>
</tr>
<tr>
<td>Team 24</td>
<td>0.555</td>
<td>0.352</td>
<td>0.431</td>
<td>VSM candidate ranking</td>
</tr>
<tr>
<td>Team 25</td>
<td>0.619</td>
<td>0.309</td>
<td>0.412</td>
<td>unsupervised with restricted information</td>
</tr>
<tr>
<td>Team 26</td>
<td>0.434</td>
<td>0.241</td>
<td>0.310</td>
<td>unsupervised; clustering based</td>
</tr>
<tr>
<td>Team 27</td>
<td>0.252</td>
<td>0.170</td>
<td>0.203</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Table 4.17: Results for queries targeting GPE (Geographical-Political) entities.

<table>
<thead>
<tr>
<th>ID</th>
<th>µAVG</th>
<th>B³ P</th>
<th>B³ R</th>
<th>B³ F1</th>
<th>B³⁺ P</th>
<th>B³⁺ R</th>
<th>B³⁺ F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>run #1</td>
<td>0.756</td>
<td>0.966</td>
<td>0.733</td>
<td>0.833</td>
<td>0.739</td>
<td>0.624</td>
<td><strong>0.677</strong></td>
</tr>
<tr>
<td>run #2</td>
<td>0.663</td>
<td>0.930</td>
<td>0.581</td>
<td>0.716</td>
<td>0.630</td>
<td>0.461</td>
<td>0.532</td>
</tr>
<tr>
<td>run #3</td>
<td>0.737</td>
<td>0.955</td>
<td>0.724</td>
<td>0.824</td>
<td>0.721</td>
<td>0.605</td>
<td>0.658</td>
</tr>
<tr>
<td>run #4</td>
<td>0.757</td>
<td>0.942</td>
<td>0.684</td>
<td>0.792</td>
<td>0.723</td>
<td>0.595</td>
<td>0.653</td>
</tr>
<tr>
<td>run #5</td>
<td>0.513</td>
<td>0.949</td>
<td>0.412</td>
<td>0.575</td>
<td>0.491</td>
<td>0.272</td>
<td>0.350</td>
</tr>
<tr>
<td>run #6</td>
<td>0.553</td>
<td>0.963</td>
<td>0.508</td>
<td>0.665</td>
<td>0.539</td>
<td>0.359</td>
<td>0.431</td>
</tr>
<tr>
<td>run #7</td>
<td>0.624</td>
<td>0.941</td>
<td>0.536</td>
<td>0.683</td>
<td>0.596</td>
<td>0.424</td>
<td>0.495</td>
</tr>
<tr>
<td>run #8</td>
<td>0.704</td>
<td><strong>0.968</strong></td>
<td>0.649</td>
<td>0.777</td>
<td>0.687</td>
<td>0.531</td>
<td>0.599</td>
</tr>
<tr>
<td>run #9</td>
<td>0.737</td>
<td>0.932</td>
<td>0.681</td>
<td>0.787</td>
<td>0.698</td>
<td>0.590</td>
<td>0.640</td>
</tr>
</tbody>
</table>
Table 4.18: Results for the queries targeting ORG (Organization) entities.

<table>
<thead>
<tr>
<th>ID</th>
<th>(\mu AVG)</th>
<th>B³ P</th>
<th>B³ R</th>
<th>B³ F1</th>
<th>B³+ P</th>
<th>B³+ R</th>
<th>B³+ F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>run #1</td>
<td>0.762</td>
<td>0.772</td>
<td>0.477</td>
<td>0.589</td>
<td>0.604</td>
<td>0.374</td>
<td>0.462</td>
</tr>
<tr>
<td>run #2</td>
<td>0.762</td>
<td>0.693</td>
<td>0.443</td>
<td>0.541</td>
<td>0.533</td>
<td>0.333</td>
<td>0.410</td>
</tr>
<tr>
<td>run #3</td>
<td>0.773</td>
<td>0.754</td>
<td>0.476</td>
<td>0.583</td>
<td>0.597</td>
<td>0.378</td>
<td>0.463</td>
</tr>
<tr>
<td>run #4</td>
<td>0.746</td>
<td>0.711</td>
<td>0.457</td>
<td>0.557</td>
<td>0.536</td>
<td>0.348</td>
<td>0.422</td>
</tr>
<tr>
<td>run #5</td>
<td>0.763</td>
<td></td>
<td></td>
<td></td>
<td>0.886</td>
<td></td>
<td></td>
</tr>
<tr>
<td>run #6</td>
<td>0.745</td>
<td>0.846</td>
<td>0.503</td>
<td>0.631</td>
<td>0.636</td>
<td>0.390</td>
<td>0.483</td>
</tr>
<tr>
<td>run #7</td>
<td>0.736</td>
<td>0.874</td>
<td>0.476</td>
<td>0.616</td>
<td>0.656</td>
<td>0.358</td>
<td>0.464</td>
</tr>
<tr>
<td>run #8</td>
<td>0.783</td>
<td>0.807</td>
<td>0.468</td>
<td>0.592</td>
<td>0.640</td>
<td>0.373</td>
<td>0.472</td>
</tr>
<tr>
<td>run #9</td>
<td>0.698</td>
<td>0.755</td>
<td>0.488</td>
<td>0.593</td>
<td>0.529</td>
<td>0.355</td>
<td>0.425</td>
</tr>
</tbody>
</table>

Table 4.19: Results for queries targeting PER (Person) entities.

<table>
<thead>
<tr>
<th>ID</th>
<th>(\mu AVG)</th>
<th>B³ P</th>
<th>B³ R</th>
<th>B³ F1</th>
<th>B³+ P</th>
<th>B³+ R</th>
<th>B³+ F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>run #1</td>
<td>0.691</td>
<td>0.856</td>
<td>0.559</td>
<td>0.676</td>
<td>0.604</td>
<td>0.428</td>
<td>0.501</td>
</tr>
<tr>
<td>run #2</td>
<td>0.636</td>
<td>0.824</td>
<td>0.511</td>
<td>0.630</td>
<td>0.530</td>
<td>0.357</td>
<td>0.427</td>
</tr>
<tr>
<td>run #3</td>
<td>0.669</td>
<td>0.806</td>
<td>0.558</td>
<td>0.660</td>
<td>0.563</td>
<td>0.409</td>
<td>0.474</td>
</tr>
<tr>
<td>run #4</td>
<td>0.691</td>
<td>0.837</td>
<td>0.553</td>
<td>0.666</td>
<td>0.591</td>
<td>0.420</td>
<td>0.491</td>
</tr>
<tr>
<td>run #5</td>
<td>0.570</td>
<td></td>
<td></td>
<td></td>
<td>\textbf{0.895}</td>
<td>0.425</td>
<td>0.511</td>
</tr>
<tr>
<td>run #6</td>
<td>0.586</td>
<td>0.869</td>
<td>0.487</td>
<td>0.624</td>
<td>0.512</td>
<td>0.324</td>
<td>0.397</td>
</tr>
<tr>
<td>run #7</td>
<td>0.620</td>
<td>0.883</td>
<td>0.479</td>
<td>0.621</td>
<td>0.561</td>
<td>0.330</td>
<td>0.416</td>
</tr>
<tr>
<td>run #8</td>
<td>0.665</td>
<td>0.874</td>
<td>0.505</td>
<td>0.641</td>
<td>0.584</td>
<td>0.380</td>
<td>0.461</td>
</tr>
<tr>
<td>run #9</td>
<td>0.672</td>
<td>0.852</td>
<td>0.555</td>
<td>0.672</td>
<td>0.589</td>
<td>0.417</td>
<td>0.489</td>
</tr>
</tbody>
</table>

The achieved results for the GPE and PER focus queries (see Tables 4.17 and 4.19) show that our MFS linking method (cf. run #1) performed best achieving B-cubed+ F1 score 0.677 for GPE and 0.501 for PER focus queries. However, the MFS performed worse for the ORG (see Table 4.18) focus queries. One of the reasons for such performance of the MFS method can be that the GPE and PER entities are less ambiguous compared to the ORG entities. The results also show that the ESA linking method (cf. run #6) achieved highest B-cubed+ F1 score 0.483 (see Table 4.18) for the ORG focus queries.

In the Entity Linking task, the queries were targeting entities in three collections of documents. The Newswire collection (NW) consisting of documents from the English Gigaword Fifth Edition, the Web collection (WB) consisting of documents from various GALE web collections, and the Discussion Forum collection (DF) consisting of documents selected from the BOLT Phase 1 discussion forums source data. Tables 4.20, 4.21 and 4.22 summarize the results for queries targeting entities in these three collections.

In the three collections, the ECC method, which takes into account the context of the entities, achieved highest B-cubed precision. However, in overall, the MFS linking based submissions achieved the highest B-cubed+ F1 scores in all three submissions.
4. Knowledge Extraction and Integration with Salient Linked Entities

Table 4.20: Results for queries from the English discussion forum documents selected from the BOLT Phase 1 forum data.

<table>
<thead>
<tr>
<th>ID</th>
<th>$\mu$AVG</th>
<th>$B^3$ P</th>
<th>$B^3$ R</th>
<th>$B^3$ F1</th>
<th>$B^{3+}$ P</th>
<th>$B^{3+}$ R</th>
<th>$B^{3+}$ F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>run #1</td>
<td>0.668</td>
<td>0.934</td>
<td>0.601</td>
<td>0.732</td>
<td>0.635</td>
<td>0.469</td>
<td>0.539</td>
</tr>
<tr>
<td>run #2</td>
<td>0.578</td>
<td>0.917</td>
<td>0.483</td>
<td>0.632</td>
<td>0.537</td>
<td>0.320</td>
<td>0.401</td>
</tr>
<tr>
<td>run #3</td>
<td>0.658</td>
<td>0.884</td>
<td>0.596</td>
<td>0.712</td>
<td>0.611</td>
<td>0.452</td>
<td>0.519</td>
</tr>
<tr>
<td>run #4</td>
<td>0.662</td>
<td>0.925</td>
<td>0.562</td>
<td>0.699</td>
<td>0.624</td>
<td>0.432</td>
<td>0.511</td>
</tr>
<tr>
<td>run #5</td>
<td>0.522</td>
<td>0.953</td>
<td>0.376</td>
<td>0.539</td>
<td>0.499</td>
<td>0.220</td>
<td>0.305</td>
</tr>
<tr>
<td>run #6</td>
<td>0.539</td>
<td>0.937</td>
<td>0.460</td>
<td>0.617</td>
<td>0.503</td>
<td>0.303</td>
<td>0.378</td>
</tr>
<tr>
<td>run #7</td>
<td>0.593</td>
<td>0.939</td>
<td>0.460</td>
<td>0.617</td>
<td>0.564</td>
<td>0.325</td>
<td>0.412</td>
</tr>
<tr>
<td>run #8</td>
<td>0.603</td>
<td>0.941</td>
<td>0.526</td>
<td>0.674</td>
<td>0.572</td>
<td>0.378</td>
<td>0.456</td>
</tr>
<tr>
<td>run #9</td>
<td>0.640</td>
<td>0.919</td>
<td>0.560</td>
<td>0.696</td>
<td>0.600</td>
<td>0.427</td>
<td>0.499</td>
</tr>
</tbody>
</table>

Table 4.21: Results for the queries from the Web documents from various GALE web collections.

<table>
<thead>
<tr>
<th>ID</th>
<th>$\mu$AVG</th>
<th>$B^3$ P</th>
<th>$B^3$ R</th>
<th>$B^3$ F1</th>
<th>$B^{3+}$ P</th>
<th>$B^{3+}$ R</th>
<th>$B^{3+}$ F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>run #1</td>
<td>0.714</td>
<td>0.857</td>
<td>0.500</td>
<td>0.631</td>
<td>0.630</td>
<td>0.392</td>
<td>0.484</td>
</tr>
<tr>
<td>run #2</td>
<td>0.706</td>
<td>0.819</td>
<td>0.455</td>
<td>0.585</td>
<td>0.576</td>
<td>0.336</td>
<td>0.424</td>
</tr>
<tr>
<td>run #3</td>
<td>0.703</td>
<td>0.832</td>
<td>0.496</td>
<td>0.622</td>
<td>0.605</td>
<td>0.377</td>
<td>0.465</td>
</tr>
<tr>
<td>run #4</td>
<td>0.706</td>
<td>0.824</td>
<td>0.481</td>
<td>0.607</td>
<td>0.589</td>
<td>0.365</td>
<td>0.451</td>
</tr>
<tr>
<td>run #5</td>
<td>0.665</td>
<td>0.923</td>
<td>0.432</td>
<td>0.588</td>
<td>0.623</td>
<td>0.304</td>
<td>0.409</td>
</tr>
<tr>
<td>run #6</td>
<td>0.647</td>
<td>0.893</td>
<td>0.472</td>
<td>0.617</td>
<td>0.594</td>
<td>0.338</td>
<td>0.431</td>
</tr>
<tr>
<td>run #7</td>
<td>0.653</td>
<td>0.902</td>
<td>0.472</td>
<td>0.620</td>
<td>0.611</td>
<td>0.337</td>
<td>0.434</td>
</tr>
<tr>
<td>run #8</td>
<td>0.729</td>
<td>0.888</td>
<td>0.467</td>
<td>0.612</td>
<td>0.661</td>
<td>0.374</td>
<td>0.478</td>
</tr>
<tr>
<td>run #9</td>
<td>0.650</td>
<td>0.838</td>
<td>0.494</td>
<td>0.622</td>
<td>0.558</td>
<td>0.355</td>
<td>0.434</td>
</tr>
</tbody>
</table>

Table 4.22: Results for queries from the Newswire documents from the English Gigaword Fifth Edition.

<table>
<thead>
<tr>
<th>ID</th>
<th>$\mu$AVG</th>
<th>$B^3$ P</th>
<th>$B^3$ R</th>
<th>$B^3$ F1</th>
<th>$B^{3+}$ P</th>
<th>$B^{3+}$ R</th>
<th>$B^{3+}$ F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>run #1</td>
<td>0.788</td>
<td>0.833</td>
<td>0.622</td>
<td>0.712</td>
<td>0.672</td>
<td>0.519</td>
<td>0.586</td>
</tr>
<tr>
<td>run #2</td>
<td>0.748</td>
<td>0.762</td>
<td>0.554</td>
<td>0.641</td>
<td>0.585</td>
<td>0.446</td>
<td>0.506</td>
</tr>
<tr>
<td>run #3</td>
<td><strong>0.779</strong></td>
<td>0.822</td>
<td>0.620</td>
<td>0.707</td>
<td>0.653</td>
<td>0.511</td>
<td>0.573</td>
</tr>
<tr>
<td>run #4</td>
<td>0.786</td>
<td>0.782</td>
<td>0.603</td>
<td>0.681</td>
<td>0.631</td>
<td>0.509</td>
<td>0.563</td>
</tr>
<tr>
<td>run #5</td>
<td>0.651</td>
<td><strong>0.883</strong></td>
<td>0.459</td>
<td>0.604</td>
<td>0.576</td>
<td>0.333</td>
<td>0.422</td>
</tr>
<tr>
<td>run #6</td>
<td>0.672</td>
<td>0.872</td>
<td>0.534</td>
<td>0.662</td>
<td>0.588</td>
<td>0.399</td>
<td>0.475</td>
</tr>
<tr>
<td>run #7</td>
<td>0.701</td>
<td>0.877</td>
<td>0.532</td>
<td>0.662</td>
<td>0.628</td>
<td>0.415</td>
<td>0.500</td>
</tr>
<tr>
<td>run #8</td>
<td>0.785</td>
<td>0.853</td>
<td>0.582</td>
<td>0.692</td>
<td>0.676</td>
<td>0.485</td>
<td>0.565</td>
</tr>
<tr>
<td>run #9</td>
<td>0.761</td>
<td>0.811</td>
<td>0.619</td>
<td>0.702</td>
<td>0.632</td>
<td>0.514</td>
<td>0.567</td>
</tr>
</tbody>
</table>

Finally, we also evaluated the performance of each individual submission for queries.
4.1. Named Entity Recognition and Linking

targeting entities which are present and not present in the reference KB. For the focus queries targeting entities not in the KB, the submission #8, which combines the MFS and ESA methods achieved highest B-cubed+ F1 score 0.601. The submission #9 which is a merged submission of MFS, ECC and ESA methods, achieved highest B-cubed+ F1 score 0.595 for the focus queries targeting entities in the KB. The results are reported in Tables 4.23 and 4.24.

Table 4.23: Results for queries targeting entities which are not present in the KB.

<table>
<thead>
<tr>
<th>ID</th>
<th>(\mu AVG)</th>
<th>B³ P</th>
<th>B³ R</th>
<th>B³ F1</th>
<th>B³⁺ P</th>
<th>B³⁺ R</th>
<th>B³⁺ F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>run #1</td>
<td>0.903</td>
<td>0.813</td>
<td>0.523</td>
<td>0.636</td>
<td>0.734</td>
<td>0.487</td>
<td>0.586</td>
</tr>
<tr>
<td>run #2</td>
<td>0.953</td>
<td>0.742</td>
<td>0.478</td>
<td>0.582</td>
<td>0.710</td>
<td>0.461</td>
<td>0.559</td>
</tr>
<tr>
<td>run #3</td>
<td>0.910</td>
<td>0.788</td>
<td>0.521</td>
<td>0.627</td>
<td>0.715</td>
<td>0.489</td>
<td>0.581</td>
</tr>
<tr>
<td>run #4</td>
<td>0.885</td>
<td>0.756</td>
<td>0.477</td>
<td>0.585</td>
<td>0.663</td>
<td>0.433</td>
<td>0.524</td>
</tr>
<tr>
<td>run #5</td>
<td>0.929</td>
<td>0.888</td>
<td>0.436</td>
<td>0.585</td>
<td>0.832</td>
<td>0.404</td>
<td>0.544</td>
</tr>
<tr>
<td>run #6</td>
<td>0.835</td>
<td>0.852</td>
<td>0.495</td>
<td>0.626</td>
<td>0.717</td>
<td>0.421</td>
<td>0.531</td>
</tr>
<tr>
<td>run #7</td>
<td>0.797</td>
<td>0.878</td>
<td>0.434</td>
<td>0.581</td>
<td>0.712</td>
<td>0.345</td>
<td>0.465</td>
</tr>
<tr>
<td>run #8</td>
<td>0.942</td>
<td>0.832</td>
<td>0.507</td>
<td>0.630</td>
<td>0.788</td>
<td>0.486</td>
<td>0.601</td>
</tr>
<tr>
<td>run #9</td>
<td>0.749</td>
<td>0.782</td>
<td>0.453</td>
<td>0.574</td>
<td>0.580</td>
<td>0.345</td>
<td>0.432</td>
</tr>
</tbody>
</table>

Table 4.24: Results for queries targeting entities which are present in the KB.

<table>
<thead>
<tr>
<th>ID</th>
<th>(\mu AVG)</th>
<th>B³ P</th>
<th>B³ R</th>
<th>B³ F1</th>
<th>B³⁺ P</th>
<th>B³⁺ R</th>
<th>B³⁺ F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>run #1</td>
<td>0.597</td>
<td>0.918</td>
<td>0.659</td>
<td>0.767</td>
<td>0.585</td>
<td>0.479</td>
<td>0.527</td>
</tr>
<tr>
<td>run #2</td>
<td>0.458</td>
<td>0.888</td>
<td>0.546</td>
<td>0.676</td>
<td>0.447</td>
<td>0.325</td>
<td>0.376</td>
</tr>
<tr>
<td>run #3</td>
<td>0.572</td>
<td>0.892</td>
<td>0.654</td>
<td>0.755</td>
<td>0.561</td>
<td>0.456</td>
<td>0.503</td>
</tr>
<tr>
<td>run #4</td>
<td>0.604</td>
<td>0.902</td>
<td>0.650</td>
<td>0.755</td>
<td>0.587</td>
<td>0.486</td>
<td>0.532</td>
</tr>
<tr>
<td>run #5</td>
<td>0.340</td>
<td>0.933</td>
<td>0.420</td>
<td>0.579</td>
<td>0.325</td>
<td>0.196</td>
<td>0.244</td>
</tr>
<tr>
<td>run #6</td>
<td>0.445</td>
<td>0.934</td>
<td>0.504</td>
<td>0.655</td>
<td>0.429</td>
<td>0.304</td>
<td>0.356</td>
</tr>
<tr>
<td>run #7</td>
<td>0.540</td>
<td>0.921</td>
<td>0.554</td>
<td>0.692</td>
<td>0.512</td>
<td>0.398</td>
<td>0.448</td>
</tr>
<tr>
<td>run #8</td>
<td>0.525</td>
<td>0.934</td>
<td>0.579</td>
<td>0.715</td>
<td>0.514</td>
<td>0.388</td>
<td>0.442</td>
</tr>
<tr>
<td>run #9</td>
<td>0.666</td>
<td>0.908</td>
<td>0.688</td>
<td>0.783</td>
<td>0.636</td>
<td>0.559</td>
<td>0.595</td>
</tr>
</tbody>
</table>
4. Knowledge Extraction and Integration with Salient Linked Entities

Findings

We hereby summarize the main findings from the evaluation:

- The MFS based linking method in overall achieved the best results. It achieved best B-cubed+ F1 score from all submitted runs.
- The context based ECC linking method achieved high B-cubed precision in general, as well as for the GPE, ORG and PER focus queries.
- The context based ESA linking method achieved best B-cubed+ F1 score for ORG focus queries.
- Submissions that merge results from the MFS, ECC and ESA methods achieved best B-cubed+ F1 score for focus queries that are targeting entities present in the reference knowledge base.
- For focus queries targeting entities in discussion fora documents, web documents, or news documents, the MFS method achieved the best B-cubed+ F1 score.

4.1.5.4 Evaluation at TAC 2014

In this section, we report on our participation at the English Entity Discovery and Linking challenge at the TAC KBP 2014\(^7\). For the challenge, we developed and evaluated several approaches for entity discovery, linking and classification. For the entity discovery we evaluated the pattern-based approach described in Section 4.1.2.1. For the entity linking, we focused primarily on two most-frequent-sense approaches (see Section 4.1.2.2): i) a pure most-frequent-sense based linking and ii) a most-frequent-sense based linking enhanced with a context. This section is based on a paper [A.7] and contributions to a technical report [A.14].

We first describe the entity discovery and linking task and the process of data preparation. Next, we describe the methods that we have evaluated. Finally, we present and discuss the results from the evaluation.

Task Description

The 2014’s Entity Linking task was extended (in comparison to TAC 2013) also with full entity detection and classification. It defines the following subtasks: spotting entities in a document corpora, linking those entities with their representation in a given reference knowledge base and classifying the entities with one of the following types: PER (Person), GPE (Geo-Political Entity) and ORG (Organization). If the entity does not belong to one of these classes, then it should not be added in the list of detected entities.

During the evaluation window each participation team was given a set of 138 documents to process. The documents contained XML markup; sometimes also not valid. The participants were asked to process not just the text content of the XML markup elements, but also the values of the XML attributes. Furthermore, the organizers asked the
participants not to extract entities from quoted text (inside the quote element). The participants were also asked to adopt their systems for specific cases of entity detection, e.g. recognizing two entities, poonam and poonam8, from the content <POSTER> poonam <poonam8...@gmail.com> </POSTER>.

The systems had to output the results and provide “mention query file” containing the query id, document id, namestring of the mention, its begin and end offset. Additionally, the system is required to provide “link ID file” with information about the query id, reference knowledge base link (or NIL link), entity type (PER, ORG or GPE) and a confidence score – if available.

Data Preparation

The reference knowledge base, same as in TAC 2013, provides identification of the entities with custom identifiers. Therefore, it was necessary to perform mapping of these identifiers. We used the same approaches as we used in TAC 2013 to map the entity identifiers. For more, see the Data Preparation section from TAC 2013 (Section 4.1.5.3), which describes in detail how we realized the mapping of identifiers.

Methodology

For the EDL task we evaluated several approaches for entity mention detection, entity linking and classification. Below we list the evaluated approaches.

For entity mention detection we evaluated our method based on manually crafted lexico-syntactic patterns (Section 4.1.2.1). In addition, we have also evaluated the performance of the state of the art NER system StanfordNER [63] which we used the baseline to compare with. For StanfordNER, we used models trained based on the CoNLL 2003 [101] dataset.

For entity linking we primarily focused on the most-frequent-sense (MFS) based linking approach. The rationale behind this decision was that in our participation at the TAC 2013 challenge [A.8] our best performing submission was the one based on the most-frequent-sense approach. Thus, in TAC 2014 we have decided to run again the MFS based linking. In TAC 2014, we evaluated two variants of the MFS based linking: i) a pure MFS based linking, and ii) a MFS based linking enhanced with context. See Section 4.1.2.2 for more details on these two MFS based linking variants.

In addition to the entity linking methods listed above, we have also evaluated the performance of the Surface Form Index (SFI) based linking method which has been considered as baseline to compare with. This approach approach uses a custom entity candidate index. Note: the SFI method has been developed by Ivo Lašek. More details on the method can be found in [A.7].

Entity classification was a new task introduced in 2014 where each entity has to be classified as person, geo-political entity or organization. For this purpose we implemented a “mapping based classification” approach. In this approach, we assume an entity is classified with a DBpedia Ontology v3.9 fine-grained class and our task is to find an appropriate mapping to one of the four entity types: Person (PER), Organization (ORG), Geo-political Entities (GPE) and Miscellaneous (MISC). Therefore, we manually established mappings.
between all 537 DBpedia classes, from the DBpedia Ontology v3.9 to the four coarse
gained types. Mappings to the entity types were created according to the “TAC KBP
2014 – Entity Linking Query Development Guidelines”\textsuperscript{18}.

In several submitted submissions, we have also combined two other approaches (as
fallback) and used as baseline to compare with. A supervised classifier which uses DBpedia
as training dataset and relies on machine learning to classify the entities into one of the
four classes. We also used StanfordNER \cite{63} as an entity classifier. See \cite{A.7} for more
details on the supervised classifier and the StanfordNER classifier. \textit{Note: the supervised
classifier was developed by Ondřej Zamazal while the StanfordNER was deployed by Ivo
Lašek.}

**Submissions Description**

For the TAC KBP 2014 Entity Discovery and Linking task we have submitted three runs.
Additionally, after the submission, we we evaluated two additional runs. The descriptions
of these runs follows.

**Run #1.** This run used the pattern based approach to detect entity mentions. Each entity
mention was further linked with the Lucene index based approach. To link the entity we
submit a Lucene search query with the entity name and the first non-disambiguation page is
considered as the correct entity link. If the entity is successfully linked, then our supervised
classifier assigns the entity one of the four defined classes (PER, ORG, GPE or MISC).
If the model failed to classify into one of the four classes, we further processed the entity
mention with the StanfordNER and performed the classification. Only entity mentions
which were classified as PER, ORG or GPE were included in the output.

**Run #1v2.** This run also uses the pattern-based approach to detect entity mentions and
uses the Lucene index to link with the first non-disambiguation page. The main purpose of
this run is to evaluate the supervised classifier. To this end, in this run we used the manual
classification approach. Since the manual classification requires DBpedia Ontology classes,
the most specific DBpedia Ontology class as returned by the Entityclassifier.eu NER was
used to map to one of the three required classes. If we failed to classify the entity, then
the StanfordNER was used to perform the classification.

**Run #1v3.** This run also uses the pattern-based approach to detect entities. The main
purpose of this run is to evaluate the quality of the Lucene-based linking approach used
in run#1, which as a correct link considers the first \textit{non-disambiguation} page. Therefore,
in this run, as the correct entity link we did not skip the disambiguation pages, and the
Wikipedia page with the highest rank in the Lucene index was considered as correct.
For classification, this run uses the supervised classifier together with a fallback to the
StanfordNER classifier.

**Run #2.** This run uses StanfordNER to extract the entity mentions. Further, it uses SFI
approach to perform entity linking and StanfordNER for the entity classification.

\textsuperscript{18}TAC 2014 EDL Query Development Guidelines - \url{http://nlp.cs.rpi.edu/kbp/2014/
annotation.html}
Run #3. This run uses the pattern-based approach to detect entity mentions. For this run we executed a Lucene based entity linking, which considers also the context around the entity when choosing the right candidate. In this run, for classification we used the supervised classifier with a fallback to the StanfordNER classifier.

Evaluation Metrics

The TAC 2014 KBP Entity Discovery and Linking challenge evaluated the performance of the entity detection, linking, classification and clustering. Below we provide brief description of the evaluation metrics.

Strong Mention Match. A micro-averaged metric for evaluation of the entity mentions. The begin and the end offsets of the entity must exactly match with the ground-truth to be counted as correct.

Strong Typed Mention Match. A micro-averaged metric for evaluation of the entity detection and classification. In addition to the begin and end offsets, also the type must match with the ground-truth to be counted as correct.

Strong All Match. A micro-averaged metric for evaluation of the entity linking. A mention is counted as correct if the link (KB link or NIL link) matches the ground-truth link.

Mention CEAF. A metric for evaluation of the entity clustering. It is based on a one-to-one alignment between system and ground-truth clusters (KB and NIL). It computes the optimal mapping based on the overlap between system-gold pairs. The entity mention offsets must match the ground-truth spans and incorrect matches affects the precision and the recall.

Results

We report all four metrics for each of our main three submissions (run #1-3) and the results of the two additional runs (run #1v2 and run #1v3). Additionally, we also report on the results for the teams at rank 1 and 10 (20 teams have participated at the challenge). Unfortunately, the organizers do not provide the results for the other teams.

Table 4.25 summarizes the results from the evaluation of the entity mention detection. The highest F1-score 0.595 was achieved by the StanfordNER (run #2) followed by the submission which uses manually crafted lexico-syntactic patterns, which achieved 0.411 F1. The reason for the lower performance of the pattern based approach is primarily due to the fact that the lexico-syntactic patterns were not optimized/defined for the TAC content. Note that our goal with this experiment was to evaluate the performance of the entity spotting approach which uses lexico-syntactic patterns on TAC content. In our future work, we will work on automatic learning of such lexico-syntactic patterns and their optimization for a domain specific content.

The results from the evaluation of the entity linking are summarized in Table 4.26. It can be observed that the most-frequent-sense approach, which uses the surface form
4. Knowledge Extraction and Integration with Salient Linked Entities

Table 4.25: Results from the entity mention detection evaluation - Strong Mention Match metric.

<table>
<thead>
<tr>
<th>Id</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>run #1</td>
<td>0.383</td>
<td>0.433</td>
<td>0.407</td>
</tr>
<tr>
<td>run #2</td>
<td>0.589</td>
<td>0.602</td>
<td><strong>0.595</strong></td>
</tr>
<tr>
<td>run #3</td>
<td>0.390</td>
<td>0.434</td>
<td>0.411</td>
</tr>
<tr>
<td>run #1v2</td>
<td>0.408</td>
<td>0.212</td>
<td>0.279</td>
</tr>
<tr>
<td>run #1v3</td>
<td>0.388</td>
<td>0.414</td>
<td>0.400</td>
</tr>
<tr>
<td>Team at rank #1</td>
<td>0.844</td>
<td>0.719</td>
<td>0.776</td>
</tr>
<tr>
<td>Team at rank #10</td>
<td>0.593</td>
<td>0.715</td>
<td>0.648</td>
</tr>
</tbody>
</table>

index (run #2) achieved highest F1-score 0.369, while second best results were achieved by the Lucene index 0.269 (run #1). Our assumption for the poorer performance of the run using the Lucene index might be due to i) the poor performance of preceding pattern-based entity spotting approach (0.407 F1 compared to 0.595 of the StanfordNER), and/or ii) our dated Lucene index, created from a Wikipedia snapshot as of 8/9/2012. For the most-frequent-sense approach, which uses the surface form index (run #2) we used more recent dataset based on Wikipedia snapshot as of 4/6/2013.

The results from the evaluation also show that run #1 which skips the disambiguation pages when performing linking achieved better results than the run #v3, which does not skip the disambiguation pages.

Table 4.26: Results from entity linking evaluation - Strong All Match metric.

<table>
<thead>
<tr>
<th>Id</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>run #1</td>
<td>0.241</td>
<td>0.272</td>
<td>0.255</td>
</tr>
<tr>
<td>run #2</td>
<td>0.365</td>
<td>0.373</td>
<td><strong>0.369</strong></td>
</tr>
<tr>
<td>run #3</td>
<td>0.221</td>
<td>0.246</td>
<td>0.232</td>
</tr>
<tr>
<td>run #1v2</td>
<td>0.258</td>
<td>0.134</td>
<td>0.176</td>
</tr>
<tr>
<td>run #1v3</td>
<td>0.261</td>
<td>0.278</td>
<td>0.269</td>
</tr>
<tr>
<td>Team at rank #1</td>
<td>0.691</td>
<td>0.656</td>
<td>0.673</td>
</tr>
<tr>
<td>Team at rank #10</td>
<td>0.492</td>
<td>0.545</td>
<td>0.517</td>
</tr>
</tbody>
</table>

Table 4.27 summarizes the results from the evaluation of entity classification. The highest F1-score 0.555 achieved the submission which relies on StanfordNER classifier. The results also show that the submission #1 which uses the supervised classifier achieved better F1-score 0.338, compared to the submission #1v2 based on the manual mappings. On the other hand, the manual mappings based submission achieved higher precision 0.368 than the supervised model 0.319. One of the reasons for the low performance of the submission with manual mappings is that the mappings obviously have low coverage (i.e.
0.191 F1-score). In our future work, we will extend the mappings so that we consider more information, such as the entity category in DBpedia (i.e. `dcterms:subject`) or the list of predicates used to describe the entity.

Table 4.27: Results from the entity classification evaluation - Strong Typed Mention Match metric.

<table>
<thead>
<tr>
<th>Id</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>run #1</td>
<td>0.319</td>
<td>0.361</td>
<td>0.338</td>
</tr>
<tr>
<td>run #2</td>
<td>0.550</td>
<td>0.561</td>
<td><strong>0.555</strong></td>
</tr>
<tr>
<td>run #3</td>
<td>0.314</td>
<td>0.350</td>
<td>0.331</td>
</tr>
<tr>
<td>run #1v2</td>
<td>0.368</td>
<td>0.191</td>
<td>0.252</td>
</tr>
<tr>
<td>run #1v3</td>
<td>0.338</td>
<td>0.360</td>
<td>0.348</td>
</tr>
<tr>
<td>Team at rank #1</td>
<td>0.769</td>
<td>0.729</td>
<td>0.749</td>
</tr>
<tr>
<td>Team at rank #10</td>
<td>0.516</td>
<td>0.717</td>
<td>0.600</td>
</tr>
</tbody>
</table>

Table 4.28 presents the results from the clustering evaluation. The best F1-score for the CEAF clustering metric was achieved by the run #2 0.429, which uses the method based on surface form index linking and the “exact name” NIL clustering technique. Second best F1-score 0.351 was achieved by submission #1, which relies on the most frequent sense approach realized with a Lucene index. Note that our main focus in this work was on the actual evaluation of the NER methods integrated in Entityclassifier.eu and not on the clustering step.

Table 4.28: Results from clustering evaluation - Mention CEAF metric.

<table>
<thead>
<tr>
<th>Id</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>run #1</td>
<td>0.331</td>
<td>0.374</td>
<td>0.351</td>
</tr>
<tr>
<td>run #2</td>
<td>0.425</td>
<td>0.434</td>
<td><strong>0.429</strong></td>
</tr>
<tr>
<td>run #3</td>
<td>0.330</td>
<td>0.368</td>
<td>0.348</td>
</tr>
<tr>
<td>run #1v2</td>
<td>0.361</td>
<td>0.188</td>
<td>0.247</td>
</tr>
<tr>
<td>run #1v3</td>
<td>0.333</td>
<td>0.355</td>
<td>0.344</td>
</tr>
<tr>
<td>Team at rank #1</td>
<td>0.750</td>
<td>0.712</td>
<td>0.730</td>
</tr>
<tr>
<td>Team at rank #10</td>
<td>0.480</td>
<td>0.670</td>
<td>0.559</td>
</tr>
</tbody>
</table>

**Lessons Learned**

We hereby summarize the lessons learned from the evaluation.
4. Knowledge Extraction and Integration with Salient Linked Entities

- **Accurate entity mention detection.** Since incorrectly spotted entity directly influences the entity linking and classification, the mention spotting is a crucial step. Therefore, in our future we will focus our efforts on developing more precise entity mention detection methods.

- **Most-frequent-sense or context based entity linking.** We evaluated also more sophisticated approaches to entity linking based on their context and co-occurrences with other entities. This approach usually performs better for rare meanings of entities. However, for the general case of TAC 2014 KBP Entity Discovery and Linking dataset the most-frequent-sense method provided better results.

- **Learning classification from knowledge graphs.** The results from the evaluation shows that open knowledge graph data is mature enough and can be also useful for learning entity classification. It would be interesting to see the impact on the performance if also other knowledge graphs, such as YAGO, are considered. This is task for our future work.

4.1.5.5 Case Study: Concept Detection in TRECVID ASR Transcripts

Although a video is a multi-modal signal consisting of visual content, audio and text (i.e. transcripts or subtitles), most recent works on concept detection have been primarily focused on exploiting the video content only. In this section, we describe a case study in which we applied a developed entity linking method on a video transcripts with the goal to improve the performance of concept detection in videos. In particular, we employed the Explicit Semantic Analysis (ESA) based entity linking method (Section 4.1.2.2) to process ASR (Automatic Speech Recognition) transcripts and link relevant concepts. We process each video transcript and computed relatedness value between the text fragment and each concept. These result vectors were further used in a fusion process and merged with the results from the video concept detection. This case study is based on data provided by TRECVID, which is an international benchmarking activity for the multimedia analysis community, where concept detection in videos is in its focus. The content of this section is based on a book chapter [A.11] and contributions to a technical report [A.15]. Note: in this work the author only contributed by processing the video transcripts and linking the concepts. The author has not contributed to the process of fusion of the results from i) processing the video transcripts using our approach, and the results from the ii) concept detection in videos (processed by the other co-authors). In the text below, we first describe the concept detection task. Next, we describe the dataset and the experiment setup. Finally, we briefly discuss the results from the experiments and the contribution of our work.

**Concept Detection in Video Transcripts**

4.1. Named Entity Recognition and Linking

The problem can be cast as a conventional document categorization problem, where the target concepts (accompanied with short descriptions) correspond to classes and the ASR fragments are the documents to be categorized (see Example 4.1.1). The ultimate goal is to estimate the relatedness between each concept and a target ASR fragment.

There is a large body of research on text categorization; for an extensive review of selected approaches please refer to [102]. Many common algorithms are based on the bag of words representation, Term Frequency - Inverse Document Frequency (TF-IDF) and cosine similarity matching [103].

Example 4.1.1. ASR text: “...looming clouds of smoke and fireballs were visible, possibly on the islands of the West a New Jersey Turnpike intention was brought down.”, target classes: Explosion-Fire, Basketball, Car-Racing

The main problem with the application of text categorization on the concept detection in video transcripts is the sparsity of the input data. Unlike in a typical text categorization setting, both input texts (concept description and ASR fragments) tend to be very short. Therefore, in our work, we adopted the Explicit Semantic Analysis (ESA) [91] algorithm for the purpose of concept detection. We used the same ESA based approached that we adopted for entity linking (see Section 4.1.2.2). Since ESA estimates relatedness between two text fragments (or two words), it can be easily adapted for text classification. Considering that each concept has a textual description, we can use ESA to estimate the semantic similarity between the document in question (ASR fragment) and each concept (i.e. its description).

While there are other Wikipedia-based algorithms, such as Wikipedia Link Measure (WLM) [104], these are less suitable for the intended fusion setup, since they compute similarity only between individual words (or Wikipedia articles), while ESA naturally handles computation of similarity between words as well as texts. Out of all Wikipedia-based word similarity/relatedness algorithms, ESA has the highest amount of a follow-up applied research in various areas of information retrieval, including cross-language information retrieval. In image and video processing, ESA was used in supporting the task of automatic image tagging [105] as well as video retrieval [106].

Dataset

In this work, we used the TRECVID 2012 Semantic indexing (SIN) task dataset [107]. The dataset consists of ASR transcripts associated with videos [108]. It consists of 19,861 videos (>600 hours) and 8,262 videos (>200 hours) for training and testing, respectively. For our part of the work (i.e. linking concepts to ASR fragments) we only considered the ASR transcripts. ASR transcripts are provided only for 14,507 training videos and 5,587 testing videos; the rest of the videos do not include speech. The ASR transcripts are derived from short videos collected from Internet archives, and add up to more than 400K video shots in total.
4. Knowledge Extraction and Integration with Salient Linked Entities

Experiment setup

In our work, we processed 34 concepts of the TRECVID SIN task. Most concepts are described with one sentence (e.g., “Shots of an airplane”, “One or more people singing”), and the remaining few concepts have a somewhat longer description (about 2-5 sentences). The objective is to detect these concepts in non-annotated video transcripts.

For processing of the ASR transcripts we used the ESA algorithm\(^\text{21}\). The input for ESA comprises the ASR fragment and the description document for each target concept. The result of the computation – the relatedness value – is used in the fusion process as the degree of confidence.

<table>
<thead>
<tr>
<th>Example 4.1.2. ASR text:</th>
<th>“...looming clouds of smoke and fireballs were visible, possibly on the islands of the West a New Jersey Turnpike intention was brought down.”.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concepts with their textual description:</td>
<td>Car-Racing – “Shot of scenes at car races.”, Explosion-Fire – “Shots of an explosion or a fire.”, and Basketball – “One or more people playing basketball.”.</td>
</tr>
<tr>
<td>The text similarity vector between the ASR text and each individual concept is</td>
<td>( \tilde{\text{sim}}(\text{ASRText}, [\text{car, fire, basketball}]) = [0.04912, 0.0814, 0.0379] ).</td>
</tr>
</tbody>
</table>

As presented in Example 4.1.2, computed relatedness values using the ESA algorithm between the fragment text and each concept description document will be: 0.04912 for the “Car-Racing”, 0.0814 for the “Explosion-Fire” and 0.0379 for the “Basketball” concept. The highest confidence score is assigned to the concept “Explosion-Fire”, followed by the concepts “Car-Racing” and “Basketball”. The concept “Explosion-Fire” receives highest confidence score because in its description appears same or similar concept(s) as in the ASR fragment (i.e. smoke, fireballs).

Summary

The goal of this work was to examine if analysis of textual information associated with videos (i.e. ASR transcripts) can improve the performance of the task of concept detection in videos. According to the results from the experiments, combination of i) the results from processing the video transcripts, and ii) the results from processing the videos, gives better results compared to the baseline. More precisely, combination of the results from processing the textual information and the results from processing the videos gives an improvement of 36.6%. More information on the fusion process, the experiments and the results from the experiments can be found in our book chapter [A.11] and in a technical report [A.15].

\(^{21}\)The ESAlib implementation obtained from http://ticcky.github.io/esalib/ with ESA background built from Wikipedia snapshot from 2005.
4.1.6 Achieved Impact and Use Cases

In order to measure the impact of the developed NER system Entityclassifier.eu, we collected basic information for the end users. The information was collected via an online form used for requesting an API key. We asked the users to provide the name of their organization, their country of origin, the language(s) they want to use and brief description of their use case.

Since January 2015, when the API key request form was launched, until August 2017 we received in total 59 requests for an API key. These 59 requests arrived from 42 unique users, from 18 different countries and 22 institutions\(^{22}\).

From the 59 API key requests, all indicated English as language of the texts to process, while 9 indicated German and 11 Dutch. While English was undoubtedly dominant, there was also reasonable interest for support of German and Dutch.

Below we present a list of use cases with the highest significance.

- (28/1/2015) Extract named entities from short news snippets.
- (29/1/2015) Execute NER in order to extract keywords from news sources.
- (26/2/2015) Perform NER on content within the H2020 FREME project\(^{23}\).
- (4/3/2015) Used by market research company to analyze and summarize answers to open questions from market research.
- (12/5/2015) Boost results from search engine using NER.
- (8/10/2015) Use in a knowledge extraction module for a decision making system.
- (6/12/2015) Tagging locations of news articles as part of a dissertation research.
- (15/12/2015) Linking university names that appear in students information table to Wikipedia pages.
- (22/12/2015) Used in project related to ontology based Information Extraction.
- (23/12/2015) Use NER to find inferences in tweets.
- (18/2/2016) Search for entities in news articles and then add extra info to the articles.
- (4/3/2016) Annotate TV programmes descriptions to improve their recommendations.
- (10/5/2016) Identify entities and concepts in short bios and messages exchanged by users in a social network.
- (16/8/2016) Annotate video files such as subtitles and description.

\(^{22}\)Note that some users did not provide the name of their organization, thus the total number of organizations is likely higher.

\(^{23}\)http://www.freme-project.eu/
4. Knowledge Extraction and Integration with Salient Linked Entities

- (22/11/2016) Categorize text in certain topics.
- (25/11/2016) Exploit in a research project to validate knowledge graph facts using text resources.
- (6/12/2016) Used in a master thesis project to annotate entities and validate the links using gamification techniques.
- (7/12/2016) Metalearning system for named entity recognition and disambiguation problems.
- (16/2/2017) Annotating a blog, adding RDFa and drawing graphs with d3.
- (26/4/2017) In a master thesis project, extract entities from news articles and use them to find structures in knowledge graphs such as DBpedia.
- (11/7/2017) Comparison between different NER systems.
- (18/7/2017) Disambiguate text for a chatbot.

In addition to the external use cases presented above, Entityclassifier.eu was initially developed for the LinkedTV project. Within the project lifetime, the system was primarily used to annotate text information related to video files. It was used to annotate subtitles and ASR transcripts of video files provided by the Dutch Sound & Vision archive and the German RBB broadcaster. See Section 4.1.5.2 for details on the validation of the system on a LinkedTV scenario.

4.1.7 Discussion and Future Work Directions

Right Type Granularity. Entityclassifier.eu uses Wikipedia to mine types which are then used to classify entities. According to the Wikipedia guidelines, the first sentence should tell the nonspecialist reader what, or who, the subject is. As a consequence, the Wikipedia editors in the first sentence of the article encode the entity type as a hypernym. Since the Wikipedia guidelines say that the first sentence is for “nonspecialist readers”, the type of the entity should be understandable by a common Wikipedia reader. In other words, it should be not too specific, neither too general. This can be interpreted as the type under which the entity is known by the general public. For example, Diego Maradona (https://en.wikipedia.org/wiki/Diego_Maradona) would be typed with its most-frequent-sense type such as “footballer” or “manager”, rather than using the very general type “person” or the very specific type “expatriate football manager in the United Arab Emirates”\(^{24}\). In our work, we mine the types from Wikipedia content and use this information to perform entity classification. Thus, we evaluated the correctness of these types with respect to the targeted entity. Evaluation of the granularity of the entity types was not in the scope of our research. In our future work, we will work on analyzing the level of specificity of the entity types.

\(^{24}\)https://goo.gl/Lt695m

106
4.1. Named Entity Recognition and Linking

**Real-time Entity Linking.** Entityclassifier.eu relies on Wikipedia for linking of the entities. In real-time (i.e. query time), every entity mention is linked to a Wikipedia page describing it and a Wikipedia URL and its corresponding DBpedia URL is assigned. Since DBpedia URLs are transparent to the Wikipedia URLs they can be easily mapped; Wikipedia uses https://en.wikipedia.org/wiki/{entity_name} URL scheme, e.g. https://en.wikipedia.org/wiki/Diego_Maradona, while DBpedia uses http://dbpedia.org/resource/{entity_name} URL scheme, e.g. https://dbpedia.org/resource/Diego_Maradona.

The benefit of the real-time entity linking is in the capability to link also new emerging entities found in the knowledge base. Our system uses live Wikipedia and thus this is possible. On the other hand, vast majority of NER systems (e.g. DBpedia Spotlight) run on a data dump. This is however a problem, since new releases of the DBpedia dataset are provided every six months and large number of entities might not be spotted since they are not present in the knowledge base. For example, during year 2017 were created additional 220,700²⁵ Wikipedia articles.

Also note that along the regular DBpedia releases (every six months), there is also a live DBpedia²⁶ instance, which in real-time processes new and updated Wikipedia articles. DBpedia live is accessible via a SPARQL endpoint²⁷ and also the existing entities (resources) are provided under dereferenceable URIs. In our future work, we plan to evaluate the performance of our system and the other related NER systems for new emerging entities.

**Real-time Type Mining and Temporal Validity.** In our system, the entity classification task is based on an information mined from Wikipedia articles. In real-time (i.e. query time), each entity mention is first link with the corresponding Wikipedia article, and in subsequent step the first sentence in the Wikipedia article is analyzed and entity classified. Since Wikipedia constantly grows, our system is able to classify also entities which were recently described in a Wikipedia page. Moreover, Wikipedia articles are constantly being updated, hence the opening sentence and the information on who/what the entity is also updated and the type becomes temporary valid. For instance, at one point of time “Donald Trump” (https://en.wikipedia.org/wiki/Donald_Trump) was a “businessman” while at another point of time he became “president”. In our work, we focus only on the real-time mining aspect, while it is part of our future work to analyze the temporal validity aspect and their validity across time. Note that we are not aware of any other system that incorporates query-time Wikipedia mining. AIDA and DBpedia Spotlight lookup the disambiguated entity in a backend database of types, which are processed from a particular knowledge base snapshot.

**Complementarity to Other Systems.** Since Entityclassifier.eu extracts the types from free text, the results are largely complementary to types returned by the other systems.

---

²⁶https://wiki.dbpedia.org/online-access/DBpediaLive
²⁷http://dbpedia-live.openlinksw.com/sparql
4. Knowledge Extraction and Integration with Salient Linked Entities

For example, DBpedia Spotlight and AIDA provide entity types which are derived from Wikipedia “infoboxes” and “categories”, respectively. Entityclassifier.eu complements this information, by mining the entity types from the Wikipedia article text. The complementary character of the different entity typing system can be utilized for a classifier fusion.

Multilinguality. NER systems, such as DBpedia Spotlight and StanfordNER, require training data for learning an entity spotting model. In fact, they require training data for each supported language. In contrast, Entityclassifier.eu does not require training data; it performs entity spotting by exploiting lexico-syntactic patterns base on Part-Of-Speech linguistic information. Extensibility to a new language requires only provisioning of a JAPE grammars and plugging in a POS tagger. Currently, Entityclassifier.eu supports English, Dutch and German. In our future work, we plan to provide support also for other languages. As part of our future work, we also plan to work on automatic learning of lexico-syntactic patterns.

4.1.8 Summary

Named Entity Recognition and Entity Linking enjoy a significant increase in popularity and usage in the last decade. The efforts within the Semantic Web and NLP communities gave birth of several NER and EL systems, however, there is still a lack of methods that do not heavily rely on training data. In this section, we presented Entityclassifier.eu, an NER system which is powered by Wikipedia and provides entity spotting, linking and classification. The entity spotting is fully unsupervised and relies on lexico-syntactic patterns. The entity classification approach is also unsupervised and it exploits text mining techniques on top of Wikipedia in order to mine entity types. Furthermore, in real-time, entities are linked with the Wikipedia and DBpedia knowledge sources and at the same time disambiguated. We executed several experiments and we evaluated the performance of the system.
4.2 Learning Entity Salience

In long documents, the number of recognized entities can be large containing also entities which are not central in the document. Identifying a subset of salient entities that play an important role in the story that the document describes can significantly help to better understand the aboutness of the document. In this section, we present a method for identification of salient entities (see Section 4.2.2.1). We apply machine learning to build a model for entity salience classification. The model is built using local information about entities derived from the document’s content, and global information about the entities from an external knowledge graph (i.e. DBpedia). The rationale for consideration of local features is that this set of features elucidate the way authors write articles and distribute entity mentions in the documents. For the global features, the rationale is that the authors write the articles with the assumption that the readers have some background knowledge about the entities and that entities popular and central in other knowledge graphs are also popular and play an important role in the document. See Step II in Section 4.2.2.2 for more information on the features.

Due to the lack of a dataset which is complete, publicly available and evaluated by humans, we exploited crowdsourcing techniques to create such dataset (see Section 4.2.3). The dataset is distributed under a free license and published in the NLP Interchange Format (NIF), which fosters interoperability and re-use. In order to validate the potential of the dataset and evaluate our methodology, we have executed several experiments (see Section 4.2.4).

The work presented in this section is based on a conference paper [A.5] and contributions to a technical report [A.14].

4.2.1 Introduction

Understanding the aboutness of Web documents is important for many Web based systems. While in the past the aboutness of documents has been primarily formulated as a problem of finding the top-K most relevant keywords in the document [109, 110] it is a recent trend that many NLP systems are moving towards understanding documents in terms of entities. Currently, there are many named entity recognition systems, however, none of the existing solutions evaluates the importance of the entity in the document. In long texts, the list of recognized entities can be very large containing also entities which are not important to the document. Identifying a subset of salient entities can significantly help to better understand the aboutness of the document.

In this section, we present a methodology for learning entity salience. The approach is entity-centric and represents the content of a provided document in terms of entities. We apply machine learning to build our model for entity salience. To train the model we leverage information about entities with local and global scope: the former describes an entity within the scope of the document and the latter describes the entity in an external knowledge base (i.e. DBpedia). Existing entity salience datasets, the Microsoft Document Aboutness (MDA) dataset [18] and the New York Times (NYT) dataset [1], face with
4. Knowledge Extraction and Integration with Salient Linked Entities

several problems; neither dataset provides the documents due to copyright restrictions and the entity annotations and salience annotations have been automatically generated and have not been manually checked (for more information see Section 2.2.2.1). Due to the lack of a dataset which is complete, publicly available and evaluated by human, we exploited crowdsourcing techniques to create such dataset. To the best of our knowledge, this is the first publicly available corpus with crowdsourced entity salience annotations which is complete, has been checked by humans and is publicly available. The corpus is created by re-using the Reuters–128 corpus [111] and published in the NLP Interchange Format (NIF) ensuring high interoperability and re-use. In our work, we use the dataset to learn entity salience.

4.2.2 Entity Salience Identification Methodology

4.2.2.1 Basic Notions

Let $e$ be a named or a common entity. We define named entity as a proper name referring to a unique entity such as location (e.g. the city Prague) or organization (e.g. the football club F.C. Milan). Further, common entities are defined as common nouns, which can refer to general names of items [112] or classes for entities (e.g., city, book or food). Further, let $D$ be a corpus of documents, and $E_d$ a set of all the named entities mentioned in a document $d \in D$.

Let $S : E_d \rightarrow C$ be a learning function which classifies an entity $e$ (named or common entity) occurring in a document $d$ into one of the finite set of salience classes $C = \{c_1, c_2, ..., c_n\}$. Each class represents a certain level of salience. In our experiments, in order to have comparable results with the related methods, we worked with a set of three salience classes {most salient, less salient, not salient}. See Section 4.2.3 for additional details.

We use RDF and the NLP Interchange Format (NIF) version 2.0\(^\text{28}\) [50] as a common data model to represent the information used in our entity salience method. The NIF format provides classes and properties to describe strings, text documents and their URI schemes. We use the $\text{nif:ContextCollection}$ concept to represent a corpus of documents $D$, the $\text{nif:Context}$ concept to represent a document $d$, and the $\text{nif:hasContext}$ predicate to link a corpus with its documents.

An entity mention is described using the concept $\text{nif:String}$, and the exact begin and end position using the $\text{nif:beginIndex}$ and $\text{nif:endIndex}$ predicates, respectively. The actual surface form of an entity mention is represented as a literal of the $\text{nif:anchorOf}$ predicate. An entity mention is at the same time disambiguated and linked using the $\text{itsrdf:taIdentRef}$ predicate from the Internationalization Tag Set (ITS) v2.0 RDF ontology\(^\text{29}\). The $\text{itsrdf:taIdentRef}$ points to a resource in a knowledge base which identifies the mentioned entity.

\(^{28}\)We refer the reader to http://prefix.cc for all prefixes used.

\(^{29}\)http://www.w3.org/TR/its20/
4.2. Learning Entity Salience

A doctor in New York City who recently returned from treating Ebola patients in Guinea became the first person in the city to test positive for the virus Thursday, setting off a search for anyone who might have come into contact with him.

**Figure 4.2: Schematic overview of the methodology for identification of salient entities.**

### 4.2.2.2 Methodology

The entity salience methodology we propose is depicted in Figure 4.2. There are two assumptions behind our methodology. First assumption is that the entity salience is function of the document’s structure. And the second assumption is that popular and central entities in external knowledge graphs could be also popular and play an important role in a given document. In order to validate our assumptions we define a methodology which consists of the following three steps: I. Entity extraction, II. Features generation, and III. Entity salience computation. The entity extraction step provides recognized and disambiguated entities. The feature computation step, derives i) features with local scope – from information available outside the scope of the document, and ii) features with global scope - from information available outside the scope of the document, from an external knowledge base. In the following text, we describe these steps in more detail.

**Step I: Entity Extraction.** In the first step, entities in a given document are recognized and disambiguated. For this purpose, we use the Entityclassifier.eu NER system (Section 4.1.3) as a generator of the initial set of candidates for salient entities. Entityclassifier.eu NER system recognises entity mentions and provides the exact location in the document by specifying the begin and the end index of the entity in the document. Entityclassifier.eu NER system also disambiguates and links recognised entities with their representation in knowledge bases. In the example in Listing 4.2 we show the results produced
from the Entityclassifier.eu system in the NIF format. The exact location of the entity mention “New York City” is covered in lines 9–10, and in line 13 the entity is disambiguated and linked with the DBpedia resource http://dbpedia.org/resource/New_York_City.

Listing 4.2: Results from named entity recognition in the NIF format.

Step II: Features generation. After an initial set of candidate salient entities is generated, follows a features computation step. We distinguish two feature categories: features with a local scope, derived from the information available within the document, and features with global scope, derived from an external entity knowledge graphs.

Features with a Local Scope. This set of features is derived only from the information available within the document. The main assumption behind this set of features is that the salience is function of the document’s structure. In other words, these features should elucidate the way authors i) write and structure articles, and ii) distribute the important entities in the document.

Some of the features, such as the 1st-begin-index and the entity-type are already provided by the NER system, while the other can be computed from the document’s content. The 1st-begin-index feature holds the positional index of the first occurrence of the entity in the document, and its value is the nif:beginIndex property in the RDF model. The feature entity-type holds the type of the entity - whether the entity in question is a named or a common entity, and it is encoded using the nif:NamedEntity and nif:CommonEntity classes. In addition to the named entities, we also consider common entities with aim to increase the coverage of our approach. The rationale behind inclusion of this feature is that common entities help better understand the aboutness of general articles and articles containing very few named entities. Extraction of common entities is already supported in some NER tools, such as DBpedia Spotlight [13] and Entityclassifier.eu [A.6]. Further, the entity-occurrences feature holds the total number of occurrences of the entity in the document. The entity-mentions feature represent the total number of entities mentioned in a given document. Finally, the unique-entities feature holds the
4.2. Learning Entity Salience

total number of unique entities occurring in a given document. The full set of local scope features is summarized in Table 4.29.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>entity-type</td>
<td>nominal</td>
<td>The type of the entity: named entity, common entity.</td>
</tr>
<tr>
<td>1st-begin-index</td>
<td>numeric</td>
<td>Begin index of the first occurrence of the entity in the document.</td>
</tr>
<tr>
<td>entity-occurrences</td>
<td>numeric</td>
<td>Number of mentions of the entity in the document.</td>
</tr>
<tr>
<td>entity-mentions</td>
<td>numeric</td>
<td>Total number of entity mentions in the document.</td>
</tr>
<tr>
<td>unique-entities</td>
<td>numeric</td>
<td>Number of unique entities in the document.</td>
</tr>
</tbody>
</table>

Table 4.29: Features computed from information available within the document.

Features with a Global Scope. Although, the evidence of entity salience can be derived effectively from the document content and its structure, extra information, derived from entity knowledge graphs, such as DBpedia [37], Freebase [113] or YAGO [65], can improve the accuracy of the learned entity salience model. To this end, we build a set of features derived from information available in DBpedia. There is twofold rationale for inclusion of global feature: first, the authors write the articles with the assumption that the readers have some background knowledge about the entities; and second, entities popular and central in external knowledge graphs could be also popular and play an important role in the document.

In order to compute the set of global features (see Table 4.30) we have considered the English DBpedia 2014 pagelinks dataset \(^{30}\) and the DBpedia knowledge graph. Before computation of the features we have considered cleansing of the DBpedia pagelinks dataset and removed redirect links which add noise.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>page-rank</td>
<td>numeric</td>
<td>PageRank value for the given entity.</td>
</tr>
<tr>
<td>hits</td>
<td>numeric</td>
<td>HITS score for the given entity.</td>
</tr>
<tr>
<td>in-degree</td>
<td>numeric</td>
<td>Indegree score for the given entity.</td>
</tr>
<tr>
<td>out-degree</td>
<td>numeric</td>
<td>Outdegree score for the given entity.</td>
</tr>
<tr>
<td>num-triples</td>
<td>numeric</td>
<td>Number of triples describing the entity.</td>
</tr>
<tr>
<td>num-props</td>
<td>numeric</td>
<td>Number of properties describing the entity.</td>
</tr>
<tr>
<td>object-props</td>
<td>numeric</td>
<td>Number of object properties describing the entity.</td>
</tr>
<tr>
<td>datatype-props</td>
<td>numeric</td>
<td>Number of datatype properties describing the entity.</td>
</tr>
</tbody>
</table>

Table 4.30: Features computed from DBpedia.

Based on this dataset we have computed PageRank [114], Hub and Authorities (HITS) [114], in-degree and out-degree that can be used to indicate the general importance or popularity of an entity. We used the graph analysis tool JUNG\(^{31}\) to compute PageRank and HITS\(^{32}\). More details on the PageRank and HITS computation setup can be found in the corresponding paper [A.5]. Note that the PageRank and HITS computation was implemented by Dinesh Reddy. The full list of general features is summarized in Table 4.30.

**Step III: Entity salience computation.** After recognition of the candidates for salient entities in Step I and features generation for the entities in Step II, the next step is to compute the salience information for these entities based on the generated feature set. With support of the learning function \(S : E_d \rightarrow C\) (see Section 4.2.2.1) we classify each entity from the entity candidates set \(E_d\) into one of the finite set of salience classes \(C = \{c_1, c_2, \ldots, c_n\}\). In our experiments (see Section 4.2.3), we worked with a set consisting of three salience classes \{most salient, less salient, not salient\}.

### 4.2.3 Crowdsourcing Entity Salience Annotations

As described in Section 4.2.2.2, our methodology requires training data. In the following, we describe the creation of a public dataset with entity salience annotations.

Our main goal is to provide an entity salience corpus that is complete, publicly available, and manually evaluated by humans. To this end, we take up the recently published Reuters-128 dataset [111] and crowdsourced the entity salience annotations. Reuters-128 is an English corpus, based on the well known Reuters21578\(^{33}\) corpus, it is stored in the NIF format and contains 128 economic news articles of average length of 261.04 words per document. The dataset has been initially proposed as a dataset for evaluation of NER systems and it provides information for 878 named entities with their position in the document (begin and end offset) and a URI of a DBpedia resource identifying the entity. For more information on the Reuters-128 dataset, please see [111].

Since the dataset only provides information about named entities found in the corpus, we have further extended the dataset with “common entities” using Entityclassifier.eu NER system [A.6] resulting in additional 3,551 entities. To obtain a gold standard of entity salience judgments we have used the crowdsourcing tool CrowdFlower\(^{34}\) to collect judgments from non-expert paid judges. The annotators have been given text and a highlighted entity, which has to be classified, following the approach taken by [18], into one of the following three classes:

- Most Salient - indicates that the document is mostly about entity, or entity plays a prominent role in the content of the document.

\(^{31}\)http://jung.sourceforge.net/

\(^{32}\)The code used to compute PageRank and HITS is publicly available at https://github.com/SemanticMultimedia/JungGraphMeasures

\(^{33}\)http://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html

\(^{34}\)http://www.crowdflower.com/
4.2. Learning Entity Salience

- Less Salient - entity plays an important role in some parts of the document.
- Not Salient - the document is not about the entity.

For each named and common entity in the Reuters-128 dataset, we have collected three judgments from annotators based in 15 different countries, including English-speaking countries, such as United Kingdom, Canada and United States. We have also manually created a set of test questions, which helped us to determine contributor’s trust score computed by the CrowdFlower platform. Only judgments from contributors with trust score higher than 70% have been included in the ground-truth. If the trust score of a contributor falls below 70%, all his/her judgments were disregarded. Each task consisted of 22 questions that had to be answered in order to complete the task.

Further, from different contributors we collected three judgments per question. Also, in order to collect judgments from different users, we have limited the maximum number of judgments per contributor to 352.

The crowdsourcing took 36 days, and in total we have collected 18,058 judgments from which 14,528 have been considered as “trusted” and 3,530 as “untrusted” judgments. The interannotator agreement between the annotators was 63.66%. Aggregated result is chosen based on the response with the greatest confidence, where the agreement is weighted by contributor’s trust score. Statistics for the crowdsourced dataset are presented in Table 4.31.

<table>
<thead>
<tr>
<th>Corpus</th>
<th>Reuters-128</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documents</td>
<td>128</td>
</tr>
<tr>
<td>Entity mentions</td>
<td>4,429</td>
</tr>
<tr>
<td>Unique entities</td>
<td>2,024</td>
</tr>
<tr>
<td>Entities linked to DBpedia</td>
<td>3,194</td>
</tr>
<tr>
<td>Most salient entities</td>
<td>804 (18%)</td>
</tr>
<tr>
<td>Less salient entities</td>
<td>1,750 (40%)</td>
</tr>
<tr>
<td>Not salient entities</td>
<td>1,875 (42%)</td>
</tr>
</tbody>
</table>

Table 4.31: Size metrics for the Reuters-128 entity salience dataset.

The complete crowdsourced dataset has been converted in the NIF format and published under the Creative Commons Public Domain CC0 license. This assures easy use and high interoperability of the corpus and its annotations.

4.2.4 Experiments

We have conducted several experiments in order to validate our dataset and evaluate its potential in learning entity salience. In the experiments we have addressed the following questions:

36[https://creativecommons.org/publicdomain/]
4. Knowledge Extraction and Integration with Salient Linked Entities

- What machine learning algorithm performs the best for our entity salience method?
- How accurate results gives our method for identification of salient named entities in comparison to the most related method [1]?
- What is the impact of individual and combined local and global set of features on the performance?
- How does the quality of the entity links influence the performance of learning entity salience?

For the experiments we used the complete Reuters-128 dataset and we performed ten-fold cross validation by partitioning the dataset into ten equal partitions and performing ten cross-validations while training on nine partitions and testing on one.

Further, in the experiments we consider the following three baseline methods against which we compare our method.

- **Positional Baseline.** An entity is considered as salient only if the begin index of the first occurrence in the document is within the first 100 characters. This also corresponds to a typical sentence length, which in average is around 100 characters long. We have experimented with different lengths and best results were achieved with the 100 characters length.

- **Entity Frequency Baseline.** This baseline method is learning from the frequency of the entity in the document. As a learning algorithm, for this method we have used the Random Forest (RF) tree algorithm.

- **TF-IDF Baseline.** The salience of the entity is determined according to the TD-IDF score of the entities. Similarly as the positional baseline, this can be considered as an unsupervised method.

**Experiment 1: Comparison of Various Algorithms.** In this experiment, we applied various machine learning algorithms to find the most suitable one for learning entity salience. We have experimented with five well-known machine learning algorithms: Naive Bayes (NB), Support Vector Machines (SVM) with polynomial kernel, k-Nearest Neighbor (k-NN) with euclidean distance function and k=1, C4.5 decision tree classifier and Random Forest tree classifier with maximum tree depth of 13\(^37\) and the number of trees set to 30. We have evaluated these algorithms on the two datasets, the Reuters-128 and the New York Times dataset.

The results show (see Table 4.32) that the best performing algorithm for both datasets is the Random Forest decision tree based classifier with 0.607 F1 for the Reuters-128 and 0.898 F1 for the NYT dataset. The second best performance shows the C4.5 decision tree classifier.

\(^{37}\)The tree depth 13 corresponds to the number of used features.
4.2. Learning Entity Salience

Table 4.32: Results for different learning algorithms. † - learning the model for SVM on the NYT corpus takes more than 24 hours to process.

<table>
<thead>
<tr>
<th>ML algorithm</th>
<th>Reuters-128</th>
<th></th>
<th>New York Times</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>0.518</td>
<td>0.488</td>
<td>0.391</td>
<td>0.862</td>
</tr>
<tr>
<td>SVM†</td>
<td>0.534</td>
<td>0.504</td>
<td>0.416</td>
<td>/</td>
</tr>
<tr>
<td>k-NN</td>
<td>0.566</td>
<td>0.564</td>
<td>0.565</td>
<td>0.857</td>
</tr>
<tr>
<td>C4.5</td>
<td>0.586</td>
<td>0.586</td>
<td>0.586</td>
<td>0.897</td>
</tr>
<tr>
<td>Random Forest</td>
<td>0.612</td>
<td>0.608</td>
<td>0.607</td>
<td>0.899</td>
</tr>
</tbody>
</table>

The scores reported in Table 4.32 are computed as “weighted average” for all classes. In Table 4.33 we report the scores for the “most salient” class only and compare our approach to the baseline methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Reuters-128</th>
<th></th>
<th>New York Times</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
</tr>
<tr>
<td>Positional baseline</td>
<td>0.518</td>
<td>0.488</td>
<td>0.391</td>
<td>0.620</td>
</tr>
<tr>
<td>Entity frequency baseline</td>
<td>0.437</td>
<td>0.133</td>
<td>0.204</td>
<td>0.706</td>
</tr>
<tr>
<td>TF-IDF baseline</td>
<td>0.407</td>
<td>0.433</td>
<td>0.420</td>
<td>0.501</td>
</tr>
<tr>
<td>Our with Random forest</td>
<td>0.693</td>
<td>0.516</td>
<td>0.592</td>
<td>0.611</td>
</tr>
</tbody>
</table>

Table 4.33: Evaluation results for different baseline methods for the “most salient” class.

The results show that our model outperforms all other considered baseline methods. For the Reuters–128 dataset our model based on the Random Forest algorithm achieves 0.592 F1, while the positional baseline 0.391 F1 and the entity frequency baseline 0.204 F1. Similarly, for the NYT dataset our method achieves 0.620 F1, while positional baseline 0.369 F1, the entity frequency baseline 0.426 F1 and the TF-IDF baseline 0.426 F1. In other words, the entity frequency baseline, which is very close to a typical keyword extraction approach, is improved by our model by 290% for the Reuters–128 dataset and by 45% for the NYT dataset.

It can be also observed that all the methods performed better on the NYT dataset than on the Reuters-128 dataset. One reason for such results may be the way how the datasets...
4. Knowledge Extraction and Integration with Salient Linked Entities

were created; while the Reuters-128 dataset was created by human annotators, the NYT
dataset was created automatically, by considering each entity which occurs in the article’s
abstract as salient entity. Another impact on the results may have the domain of the
dataset. While the NYT contains newspaper articles from various domains, the Reuters-
dataset contains economic news article, which might be more difficult for learning
entity salience.

**Experiment 2: Comparison to the NYT entity salience approach [1]**. A particular
method that relates to our work is the one proposed in [1]. The authors propose model
for learning entity salience using features derived from a coreference resolution system
and leveraging background information. In order to have comparable results, we have
evaluated our approach on the NYT dataset and in Table 4.34 we report on the results
from our method and the one described in [1].

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our method</td>
<td>0.611</td>
<td>0.629</td>
<td>0.620</td>
</tr>
<tr>
<td>The NYT method [1]</td>
<td>0.605</td>
<td>0.635</td>
<td>0.620</td>
</tr>
</tbody>
</table>

Table 4.34: Evaluation results for our method and the NYT method [1].

According to the results in Table 4.34, our model achieved higher precision (0.611)
compared to precision reported for the NYT model (0.605). Furthermore, our model
achieved recall 0.629, and 0.620 F1, while their model, recall 0.635 and 0.620 F1. A
discussion and comparison of our approach for entity salience and the NYT approach can
be found in Section 4.2.5.

**Experiment 3: Feature Analysis**. In this experiment, we have focused on analysis of
the features and their impact on the performance. We have evaluated the contribution of
each feature with a local and global scope. The contribution of each individual feature
has been evaluated by incrementally adding new features, starting with the `begin-index`
feature from the local features set, and `page-rank` from the global features set. Table 4.35
summarizes the results from the evaluation.

The results show that the model based on the local features achieves better results
than the model based on the global features. For the Reuters–128 dataset the local fea-
tures achieve 0.592 F1 and 0.489 F1 for the global features. In comparison, for the NYT
dataset the local features achieve 0.895 F1 and 0.806 F1 the global features. It can be also
observed that a model which considers both, the local and the global features, achieves
better results than model which considers the local or the global features only. For the
Reuters–128 dataset, the improvement is around 3%, while for the NYT dataset the im-
provement is 1% only. Nevertheless, both, the global and the local features achieve suffi-
ciently good performance to be considered individually, when one or another feature family
is not available.
For the Reuters-128 dataset, the local feature `1st-begin-index` alone achieves 0.492 F1. The improvement at F1 score for each incrementally added local feature is as follows. The `entity-occurrences` 7%, `unique-entities` 4% and `entity-type` 1%. For the NYT dataset, the `entity-occurrences` improves F1 by 1%, while the rest of the features add no significant improvement. For the Reuters-128 dataset, the `page-rank` global feature individually achieves 0.394 F1. The improvement of F1 score for each incrementally added local feature is as follows. The `hits` feature 17%, the `in-degree` 5%, the `out-degree` has no significant negative influence, the `num-triples` 2%, while the `num-properties`, `num-obj-properties` and `num-datatype-properties` have insignificant negative or very low improvement. In contrast, for the NYT the `page-rank` feature individually achieves 0.806 for F1, while for each additional incrementally added feature, there is no remarkable influence on the F1 score but they influence the precision/recall. The highest impact on the precision of 0.885 has the `num-obj-properties` feature, while the `page-rank` feature has precision of 0.753.

<table>
<thead>
<tr>
<th>Features</th>
<th>Reuters-128</th>
<th></th>
<th></th>
<th>New York Times</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
<td>Precision</td>
</tr>
<tr>
<td><code>1st-begin-index</code></td>
<td>0.496</td>
<td>0.493</td>
<td>0.492</td>
<td>0.863</td>
<td>0.882</td>
<td>0.865</td>
<td>0.895</td>
</tr>
<tr>
<td>+ entity-occurrences</td>
<td>0.533</td>
<td>0.531</td>
<td>0.530</td>
<td>0.895</td>
<td>0.904</td>
<td>0.894</td>
<td>0.895</td>
</tr>
<tr>
<td>+ entity-mentions</td>
<td>0.568</td>
<td>0.566</td>
<td>0.565</td>
<td>0.895</td>
<td>0.905</td>
<td>0.895</td>
<td>0.895</td>
</tr>
<tr>
<td>+ unique-entities</td>
<td>0.587</td>
<td>0.585</td>
<td>0.585</td>
<td>0.895</td>
<td>0.905</td>
<td>0.895</td>
<td>0.895</td>
</tr>
<tr>
<td>+ entity-type</td>
<td>0.596</td>
<td>0.593</td>
<td>0.592</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td>page-rank</td>
<td>0.451</td>
<td>0.455</td>
<td>0.394</td>
<td>0.753</td>
<td>0.868</td>
<td>0.806</td>
<td>0.768</td>
</tr>
<tr>
<td>+ hits</td>
<td>0.514</td>
<td>0.497</td>
<td>0.461</td>
<td>0.769</td>
<td>0.866</td>
<td>0.806</td>
<td>0.774</td>
</tr>
<tr>
<td>+ in-degree</td>
<td>0.516</td>
<td>0.504</td>
<td>0.483</td>
<td>0.769</td>
<td>0.866</td>
<td>0.806</td>
<td>0.774</td>
</tr>
<tr>
<td>+ out-degree</td>
<td>0.505</td>
<td>0.497</td>
<td>0.481</td>
<td>0.774</td>
<td>0.867</td>
<td>0.806</td>
<td>0.774</td>
</tr>
<tr>
<td>+ num-triples</td>
<td>0.520</td>
<td>0.508</td>
<td>0.492</td>
<td>0.753</td>
<td>0.868</td>
<td>0.806</td>
<td>0.753</td>
</tr>
<tr>
<td>+ num-properties</td>
<td>0.524</td>
<td>0.509</td>
<td>0.492</td>
<td>0.753</td>
<td>0.868</td>
<td>0.806</td>
<td>0.753</td>
</tr>
<tr>
<td>+ num-object-properties</td>
<td>0.522</td>
<td>0.507</td>
<td>0.490</td>
<td>0.885</td>
<td>0.868</td>
<td>0.806</td>
<td>0.885</td>
</tr>
<tr>
<td>+ num-datatype-properties</td>
<td>0.521</td>
<td>0.507</td>
<td>0.489</td>
<td>0.797</td>
<td>0.867</td>
<td>0.806</td>
<td>0.797</td>
</tr>
<tr>
<td>all combined</td>
<td><strong>0.612</strong></td>
<td><strong>0.608</strong></td>
<td><strong>0.607</strong></td>
<td><strong>0.899</strong></td>
<td><strong>0.908</strong></td>
<td><strong>0.898</strong></td>
<td><strong>0.899</strong></td>
</tr>
</tbody>
</table>

Table 4.35: Evaluation results for different features.

**Experiment 4: Impact of the Entity Linking.** The quality of the links identifying the entities is crucial for computation of different features. Based on these URIs we can compute the number of occurrences of an entity, or count the number of unique entities. Moreover, by linking the entities with an external knowledge graph we can compute various graph metrics, such as the PageRank, HITS or in/out degree.

To evaluate the impact of the quality of entity linking, we randomly created incorrect links in the dataset. Five versions of the dataset were created with different amount of
incorrectly linked entities. For this experiment, we trained the models using the Random Forest learning algorithm. The results from the experiment are presented in Table 4.36.

<table>
<thead>
<tr>
<th>Portion of incorrect links</th>
<th>Precision</th>
<th>Recall</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>all correct</td>
<td>0.612</td>
<td>0.608</td>
<td>0.607</td>
</tr>
<tr>
<td>1/10</td>
<td>0.597</td>
<td>0.592</td>
<td>0.592  (-2.47%)</td>
</tr>
<tr>
<td>1/5</td>
<td>0.588</td>
<td>0.583</td>
<td>0.582  (-4.12%)</td>
</tr>
<tr>
<td>1/4</td>
<td>0.587</td>
<td>0.583</td>
<td>0.582  (-4.12%)</td>
</tr>
<tr>
<td>1/3</td>
<td>0.571</td>
<td>0.566</td>
<td>0.565  (-6.92%)</td>
</tr>
<tr>
<td>1/2</td>
<td>0.560</td>
<td>0.555</td>
<td>0.553  (-8.90%)</td>
</tr>
</tbody>
</table>

Table 4.36: Results for different portions of incorrectly linked entities.

The results from the experiment show that the entity salience learning is influenced by the quality of the entity linking where 10% of incorrectly linked entities results in 2.47% decrease, while 20-25% of incorrectly linked entities result in 4.12% decrease of the accuracy of the salience learning. It can be also observed, that even with 50% of incorrectly linked entities, the learning accuracy still shows promising results (F1=0.553). This shows that the features which are not dependent on the entity links (1st-begin-index, entity-mentions, entity-type) can balance the incorrectly linked entities.

4.2.5 Discussion

Identification of salient entities is a relatively new research problem and there are two key contributions of our work in the context of learning entity salience: i) a method for learning entity salience, and ii) a crowdsourced dataset with entity salience annotations.

Learning Entity Salience. The authors in [18] develop a supervised model for learning document aboutness through identification of salient entities. Nevertheless, there are two key differences between our proposed method and theirs. First, for learning we use data labeled by humans, while in their work they used behavioral signals to collect salience annotations. And second, in addition to the features derived from information available with the document, we consider also features derived from graph analysis of an external entity knowledge graph.

In [80] the authors represent the aboutness using words and phrases that best reflect the central topics of a document, while in our approach we represent the document aboutness in terms of entities. Moreover, in their work the training data is generated from implicit user feedback available in search engine click logs, while in our work through crowdsourcing we collected labeled data with salience entity annotations.

Representing aboutness of microposts (tweets) by linking the tweets with semantically related Wikipedia pages has been proposed in [81]. The aboutness of tweets has been
4.2. Learning Entity Salience

defined as determining a set of concepts “contained in, meant by, or relevant” to the tweet. In contrast, in our work we do not aim at identification of all entities, but only the most central one. Moreover, we focus on identification of the entities which play an important role in the document, and identification of “related entities” is out the scope of our research.

A particular method that relates to our work is the one proposed in [1]. The authors propose model for learning entity salience using features derived from a coreference resolution system and leveraging background information. Our entity salience model differs in several aspects compared to theirs. First, we use more specific local features for positioning the entity in the document, for instance, we rely on the begin index of the first entity occurrence in the document, while in their model they consider the sentence index of the entity occurrence. Next, for learning, we use data labeled by humans, while in their work they used automatically derived salience information. Finally, in our work we differently incorporate background information about the entities found external knowledge graphs.

When computing the graph metrics, such as the PageRank or HITS, we use the whole external knowledge entity graph (i.e. DBpedia), while in their work, the graph metrics are computed only from graph consisting of entities occurring in the document itself. Also, in our work, we consider several graph metrics such as the PageRank, HITS, in-degree, out degree, while in their work only PageRank is considered. We have also evaluated our model (for the “salient class”) on the same dataset used in [1] (see Experiment #2 in Section 4.2.4), and our model achieved higher precision (0.611) compared to precision reported for their model (0.605). Furthermore, our model achieved recall 0.629, and 0.620 F1, while their model, recall 0.635 and 0.620 F1.

Entity Salience Datasets. There are two entity salience datasets that have been published recently, the “Microsoft Document Aboutness” (MDA) [18] and the “New York Times” (NYT) [1] dataset. While both datasets, the MDA and NYT, do not provide the documents due to copyright restrictions, our dataset is publicly available, including the documents. Also, in our dataset, the entity annotations have been checked by human, while in MDA and NYT they have been generated automatically using a proprietary NER and have not been checked by human. Moreover, the salience annotations in the NYT dataset have have been automatically generated by aligning the entities in the abstract and the document under the assumption that every entity which occurs in the abstract is salient, which is questionable. Overall, the design of these datasets does not foster their straightforward use and efficient development of methods for identification of salient entities.

4.2.6 Summary

In the recent years, named entity recognition systems gained great popularity on the Web. However, these systems, at large, do not evaluate the actual importance of the recognized entities in the documents. In this section, we presented a methodology for identification of salient entities. For a given document, a set entities which play an important role in the document are discovered. We use supervised machine learning to build our model for entity salience. Our model exploits information about entities derived from the document’s content and information from entity knowledge graphs. Since we tackle relatively
new research problem, a complete, publicly available and evaluated by human dataset for
learning and evaluation of entity salience is practically non-existent. To this end, through
crowdsourcing we created the first publicly available and complete entity salience dataset.
We validated and evaluated the method on this newly created dataset and the results show
that our method outperforms the baseline methods. The results also show that the fea-
tures with local scope show stronger salience cues than the set of features derived from the
entity knowledge graph. Nevertheless, the global and local features combined show better
performance.

In our future work, we want to extend the features set with additional features derived
by deep analysis of entity knowledge graphs from the LOD cloud. We also plan to adapt
and apply our entity salience model on different domains such as microposts, video subtitles
and music lyrics. Last but not least, we would like to exploit also other knowledge bases,
such as Wikipedia or the Wikilinks\textsuperscript{38} corpus, for learning entity salience.

\textsuperscript{38}https://code.google.com/archive/p/wiki-links
4.3 Evaluation Framework for Named Entity Recognition

While NER systems are gaining their popularity, there is yet no oversight on their performance in general, and their performance in specific domains. Currently, there are three elementary tasks that an NER and EL system performs: i) entity spotting, ii) entity linking (linking entity mentions with their instances in a knowledge base), and iii) entity classification. As knowledge for disambiguation and linking entities usually use large scale knowledge bases, such as DBpedia and Wikipedia. Classes for entity classification are usually assigned from a taxonomy, which makes the output of the NER system comparable with other systems.

In order to enable comparison of different NER systems, we developed an evaluation framework for benchmarking NER systems (see Section 4.3.3). It is developed as a stand-alone project on top of the GATE text engineering framework and it is primarily developed for off-line evaluation of NER systems. Since different NER systems might perform better on one and worse on another domain, we have also developed two annotated datasets with entities, the News and the Tweets dataset (see Section 4.3.2). The Tweets dataset, consists of very large number of short texts (tweets), while the News dataset consists of standard-length news articles. Entities recognized in the original datasets were enriched with new annotations (see Section 4.3.2) – a link to Wikipedia and the most specific type from the DBpedia Ontology, which was selected as the set of types for the fine-grained classification. We have found this choice natural, due to its wide adoption and the fact that mappings between its classes and the proprietary types output by many entity classification systems, such as OpenCalais, AlchemyAPI and Zemanta, are already provided by the NERD ontology. The framework is also supported with three plugins, which have been developed for the GATE text engineering platform. The datasets are supplemented by plugins for their import to the GATE framework and a DBpedia Ontology-aware plugin for aligning annotations created by a NER system with the ground truth. The work described in this section is based on a paper [A.9], a book chapter [A.12] and contributions to a technical report [A.13].

4.3.1 Available Datasets

Currently, there is a lack of resources for NER and entity linking evaluation, since those previously created for benchmarking of NER systems cannot be directly used. While some datasets have been recently contributed, in particular WEKEX or MSM, they do not
contain all the necessary features for automated evaluation of NER systems. The WEKEX dataset contains manual evaluation of results obtained via the NERD framework [15] from multiple common NER systems. This provides a very useful benchmark of the involved systems at the point in time when the assessment was performed. However, the design of the dataset does not foster its straightforward reuse for a new evaluation. The actual documents are not provided, entity types are not aligned nor uniquely identified with a formal ontology, entities are only classified with coarse grained types, and there is no information if the listed entity is a common or named entity. Another evaluation datasets are: the MSM challenge dataset aiming at evaluation of coarse-grained type assignment, and the NIST TAC Entity Linking contest dataset\footnote{\url{http://www.nist.gov/tac/2013/KBP/EntityLinking/}}, used of for evaluation of entity linking. Only the latter provides a comprehensive set of resources for the evaluation of NER systems (entity linking only). Unfortunately, this dataset is only available for purposes of the TAC contest. In other words, if one would like to use the TAC dataset, he/she would need to take direct participation in the challenge, which is organized ones a year. Also, when requesting the data, the user agrees not to otherwise publish, retransmit, disclose, display, copy, reproduce or redistribute the data. Moreover, after participation in the TAC evaluation, the user has to delete the data from any computer or media onto which it has been copied. Finally, none of the listed datasets support evaluation of a fine-grained classification of entities.

In this work, we take up two recently published datasets, the WEKEX dataset and the MSM Challenge 2013 dataset and extend them to fit the needs of Wikipedia/DBpedia-based entity linking and classification, creating the \textit{News} dataset, and the \textit{Tweets} dataset. The two datasets are complementary in that the WEKEX dataset consists of a small number of standard-length news articles, while the MSM datasets contains a large number of very short texts (tweets). Table 4.37 gives an overview of the size of both datasets.

<table>
<thead>
<tr>
<th></th>
<th>Documents</th>
<th>All</th>
<th>With CoNLL type</th>
<th>Ontology type</th>
<th>Wikipedia URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>News</td>
<td>10</td>
<td>588</td>
<td>580</td>
<td>367</td>
<td>440</td>
</tr>
<tr>
<td>Tweets</td>
<td>1044</td>
<td>1523</td>
<td>1523</td>
<td>1379</td>
<td>1354</td>
</tr>
</tbody>
</table>

4.3.1.1 WEKEX Dataset

The 2011 paper [15] presents an evaluation of common entity recognition systems using the NERD framework. In this evaluation there were two tasks. We created the \textit{News} dataset from the first task’s data. In this task, four participants rated the entities output by the
4.3. Evaluation Framework for Named Entity Recognition

individual systems for ten English news articles selected from the on-line archives of the BBC and The New York Times. These articles were from five different categories. For each entity, a Wikipedia link, if available, and the assigned type, were assessed.

Since each entity recognition tool recognized a slightly different set of entities, the set of all distinct entities identified by the benchmarked systems in [15] were considered for the News dataset.

A limitation of the original WEKEX dataset is the copyright restriction for the underlying textual content, the entity types are not aligned with a formal ontology, and there is no information whether the listed entity is a common or named entity. While this textual content is freely available from the BBC’s and NYTimes’s official websites, it cannot be distributed along with the annotations. The WEKEX dataset is released under the Creative Commons BY-SA 3.0 license. Links to the original content are listed on the dataset website.

4.3.1.2 MSM Challenge Dataset

The Making Sense of Microposts (MSM) Challenge 2013 dataset aimed at classifying entities in microposts (tweets). There are four entity types considered corresponding to the standard CoNLL categories: Person, Organization, Location, and Miscellaneous. The dataset is split into two parts – training and test data. Both provide annotated entities in the text. The entities in the training dataset have the types already assigned, while the test dataset is missing types. The organizers also published the goldstandard for the test dataset containing the correct entity types. The original MSM dataset is provided under the Creative Commons BY-NC-SA license.

To construct the Tweets dataset, we used the tweets in the goldstandard that contained at least one entity, resulting in 1044 tweets (1523 entities).

4.3.2 News and Tweets Evaluation Datasets

The News and Tweets datasets were created by partial reannotation and enrichment of the WEKEX and MSM datasets.

4.3.2.1 Annotation Guidelines

The WEKEX and MSM datasets were reannotated to a (nearly) common set of fields. The original version of both datasets already provided entity recognition. The annotators thus worked on the same set of entities, providing for each entity:

- URL to English Wikipedia: a URL of an article describing the entity,
- Fine-grained type: a class from the DBpedia Ontology 3.8,
4. Knowledge Extraction and Integration with Salient Linked Entities

- Coarse-grained type: a CoNLL category (only for WEKEX)\textsuperscript{51}
- Most frequent sense flag: 1 if the correct Wikipedia page is found as the first hit of Wikipedia search for the entity name, otherwise empty.

For the News dataset, we considered as entity candidates each entity output by any of the systems that generated the original annotations in the WEKEX dataset. To deal with this broader scope and lower quality, several specific annotation fields were added to the News dataset:

- Common entity: 1 if the entity is not a named entity,
- Full name: if this specific entity is a part of a full entity name, which appears in the article, then this field lists the full entity name,
- Partial: 1 if the recognized string is a part of the entity name, and this part does not appear as a full standalone reference to the entity in the document.

The entities for which the result of the annotation process was “not an entity” were removed.

The MSM dataset contains high-quality recognition of entities (with the definition of entity being narrowed to named entities), therefore entity recognition can be reused in the Tweets dataset. However, there was one problem related to entity recognition - the frequent incorrect letter casing characteristic for tweets. For the Tweets dataset, there was thus one dataset-specific field added:

- Incorrect capitalization: 1 if there is at least one letter in the entity name with incorrect case.

4.3.2.2 Annotation Process

Each entity was independently annotated by two annotators. If their annotations matched in the specific field (e.g. link to English Wikipedia), this annotation was automatically merged to the ground truth. The annotators were instructed to provide explanation if they felt unsure. When there was no match, another annotator, or in particularly spurious cases two annotators, resolved the conflict, using the explanations from the first-round annotators. The interannotator agreement after the first round (between the two annotators) for the core fields is given in Table 4.38.

The composition of annotators was as follows: one undergraduate computer science student, two graduate computer science students and one post-doc specializing on ontology alignment (all non-native English speakers). None of the authors took part in the annotation process.

\textsuperscript{51}The MSM dataset already contains this information.
Table 4.38: Interannotator agreement for the core fields. The MSM dataset already contained a coarse grained type assignment, which was not reannotated.

<table>
<thead>
<tr>
<th></th>
<th>Wikipedia URL</th>
<th>Coarse grained type</th>
<th>Fine grained type</th>
<th>Most frequent sense</th>
</tr>
</thead>
<tbody>
<tr>
<td>News</td>
<td>0.61</td>
<td>0.65</td>
<td>0.70</td>
<td>0.77</td>
</tr>
<tr>
<td>Tweets</td>
<td>0.79</td>
<td>n/a</td>
<td>0.64</td>
<td>0.86</td>
</tr>
</tbody>
</table>

4.3.3 GATE Evaluation Framework

To facilitate the use of the newly created Tweets and News datasets, we have developed three plugins for the GATE Text Engineering framework.

The NewsCorpusBuilderPR and TweetsCorpusBuilderPR plugins load the datasets into GATE. It is assumed that the NER system being benchmarked is also wrapped as GATE plugin, creating GATE annotations on entities recognized in the documents, with annotation features corresponding to the entity type and Wikipedia URL. We provide a reference implementation of such a plugin for the Entityclassifier.eu\textsuperscript{52} NER system [A.6].

Once the NER system has been run, the correct recognition of entities and the assignment of Wikipedia URLs can be evaluated by the standard GATE means. We recommend using the GATE Corpus Quality Assurance tool. The evaluation of the assigned DBpedia Ontology types needs to be performed in an ontology-aware fashion, which is not supported by this GATE tool. Consider the case, when the ground truth type for an entity is \texttt{dbo:VicePrimeMinister} and the benchmarked tool assigns \texttt{dbo:Person}. While the string-based comparison performed by the Quality Assurance tool would mark such annotation as incorrect, actually the \texttt{dbo:Person} is correct with respect to ground truth, albeit more generic.

The developed OntologyAwareFeatureDiffPR GATE plugin performs comparison of the assigned types taking into account the hierarchy of the DBpedia Ontology. The plugin also assigns a new feature \texttt{matchtype}, which has either of the following values:

- \texttt{exact match}: ground truth and NER system types match,
- \texttt{supertype}: the ground truth annotation is a super-class of the assigned class,
- \texttt{subtype}: the ground truth annotation is a sub-class of the assigned class, or
- \texttt{nomatch}: neither of the above.

In case of supertype/subtype match, the plugin also uses the \texttt{distance} feature to denote the length of the path between the subtype and supertype classes. Note that the DBpedia Ontology, seen as a taxonomy tree, does not contain cycles and the path length can be thus easily computed.

The plugin also creates a new feature \texttt{aligned-type}, which is set to the common supertype (if exists) of the fine-grained type assigned by the NER system and ground truth.

\textsuperscript{52}http://entityclassifier.eu/
4. Knowledge Extraction and Integration with Salient Linked Entities

Fine-grained type. Consider the following example. For a given entity, the ground truth annotation contains the type `dbo:VicePrimeMinister`, while the NER system assigns the type `dbo:Person`. The plugin will create the `aligned-type` feature and set it to `dbo:Person` on both the ground truth and NER system annotations. This will allow the native GATE Corpus Quality Assurance tool to evaluate this annotation as correct, while the `matchtype` feature holds the detail of the match.

4.3.4 Availability and License

The reannotated WEKEX (News) and the MSM (Tweets) datasets are available online at http://ner.vse.cz/datasets/evaluation/benchmark-datasets/. The News dataset does not contain the source texts, which need to be obtained from the BBC and NYTimes websites.

The datasets are published under the same licenses which are used for the original datasets: Creative Commons BY-SA license for News and Creative Commons BY-NC-SA license for Tweets.

The evaluation framework – the `NewsCorpusBuilderPR`, `TweetsCorpusBuilderPR` and the `OntologyAwareFeatureDiffPR` plugin, together with reference implementation of a GATE PR plugin providing annotations from the `EntityClassifier.eu` system, are available online at http://ner.vse.cz/datasets/evaluation/tools/. These plugins are provided under the GNU Lesser General Public License 3.0.

4.3.5 Evaluation Results

We used the developed evaluation framework to evaluate the performance of the EntityClassifier.eu NER system. We evaluate the performance of the system on both datasets, the Tweets and the News dataset, for the entity spotting, linking and classification. Table 4.39 summarizes the results from the evaluation on the Tweets dataset. We report on the standard precision, recall and F1 score metrics for each of the evaluated tasks. Each metric has been calculated according to two different criteria - `strict` and `lenient`. The `strict` measure considers all partially correct responses as incorrect, while the `lenient` measure considers all partially correct responses as correct.

Table 4.39: Evaluation results for the EntityClassifier.eu NER system on the Tweets benchmark dataset.

<table>
<thead>
<tr>
<th></th>
<th>Precision (strict/lenient)</th>
<th>Recall (strict/lenient)</th>
<th>F1 score (strict/lenient)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spotting</td>
<td>0.45/0.56</td>
<td>0.67/0.84</td>
<td>0.54/0.67</td>
</tr>
<tr>
<td>Linking</td>
<td>0.24/0.26</td>
<td>0.36/0.39</td>
<td>0.29/0.31</td>
</tr>
<tr>
<td>Classification</td>
<td>0.12/0.13</td>
<td>0.17/0.19</td>
<td>0.14/0.15</td>
</tr>
</tbody>
</table>
4.4. Chapter Summary

The results show that best performance has been achieved for spotting (0.54 F1 strict, 0.67 F1 lenient), followed by linking (0.29 F1 strict, 0.31 F1 lenient) and finally classification (0.14 F1 strict, 0.15 F1 lenient). In Table 4.40, we report on the results from the evaluation on the News benchmark dataset.

Table 4.40: Evaluation results for the Entityclassifier.eu NER system on the News benchmark dataset.

<table>
<thead>
<tr>
<th></th>
<th>Precision (strict/lenient)</th>
<th>Recall (strict/lenient)</th>
<th>F1 score (strict/lenient)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spotting</td>
<td>0.69/0.78</td>
<td>0.33/0.38</td>
<td>0.45/0.51</td>
</tr>
<tr>
<td>Linking</td>
<td>0.37/0.41</td>
<td>0.18/0.20</td>
<td>0.24/0.27</td>
</tr>
<tr>
<td>Classification</td>
<td>0.18/0.20</td>
<td>0.10/0.12</td>
<td>0.13/0.15</td>
</tr>
</tbody>
</table>

According to the results, Entityclassifier.eu shows worse F1 score on the News dataset than on the Tweets dataset. On the other hand, it shows better precision but lower recall on the News dataset than on the Tweets dataset.

4.3.6 Summary

While there are several ground truth datasets for evaluation of “classic” NER systems, a freely obtainable dataset for evaluation of NER and Entity Linking systems does not, to the best of our knowledge, exist (as of July, 2013). We described a framework and two complementary datasets for benchmarking NER and EL systems, which will hopefully significantly reduce the effort required to evaluate NER systems.

4.4 Chapter Summary

Nowadays, much of the content available on the Web is provided in an unstructured format. Realizing the vision of the Web of Data requires NLP approaches that enable extraction of the hidden information from unstructured data and its integration with the Linked Data space.

This chapter describes our work within the second activity of the thesis, the knowledge extraction and integration using salient named entities. We present several developed methods for entity spotting, linking and classification which have been implemented as part of the open-source NER system Entityclassifier.eu. The system enables users to choose the preferred method for entity spotting, linking and classification and select the preferred knowledge source. Our entity linking approaches are based on the most-frequent-sense and context based linking. The developed entity classification approach is designed to perform real-time entity classification with help of Wikipedia. We also presented a methodology for identification of salient entities. Via crowdsourcing we created a dataset with salience
annotations, which have been used to train a model for identification of salient entities. We also presented an evaluation framework for named entity recognition systems. We validated and evaluated the developed methods on several datasets from various domains. We also participated in several evaluation challenges and reported on the results.
Personalized Retrieval of Linked Data

Recent efforts in the Semantic Web community have been primarily focused on developing technical infrastructure and technologies for efficient Linked Data acquisition, publishing and interlinking. Nevertheless, due to the huge and diverse amount of information, the retrieval of information from the LOD cloud still demands significant amount of effort. The standard retrieval mechanisms, SPARQL and URI dereferencing, require background knowledge on the schema and have query expressivity limitations. In other words, in order to retrieve information, users can either write, usually complex, SPARQL queries, or dereference an initial set of seed URIs and iteratively follow the additional links, explore the graph and find relevant information. This, however, requires significant amount of effort from the users. This chapter describes our work within the third activity of the thesis, the personalized knowledge retrieval activity. In our work, we focused on Linked Data and developed a novel configurable method for personalised retrieval of Linked Data resources (see Section 5.2). The method recommends resources of interest from users with similar tastes. To measure the similarity between the users we introduce a novel resource semantic similarity metric, which takes into account the commonalities, the connectiviteness and informativeness of the resources.

In this work, we exploit the results from the first two activities of the thesis. The first activity provides an ontology and a dataset with semantic Web service descriptions, which has been further enriched with links to the LOD cloud using the NER system developed within the second activity of the thesis. We use the dataset in order to validate and demonstrate our method for personalised retrieval of Linked Data. We develop a resource recommendation use case for the Web services domain: recommendation of resources representing Web APIs (see Section 5.3.2). We evaluate the method on several experiments and show that the method produces more accurate, serendipitous and diverse recommendations compared to the traditional recommendation techniques (see Section 5.3.3).

Section 5.4 compares our method to the related work, discusses particular aspects of our method and presents our future work plans. Finally, Section 5.5 summarizes our achievements.

Our main contributions presented in this chapter are fivefold:
5. Personalized Retrieval of Linked Data

- A method for personalized retrieval of Linked Data, which takes into account the commonalities, the informativeness and the connectiviteness of the resources. The method (see Section 5.2) can be adapted to any dataset and domain, it does not require pre-processing of the datasets, it exploits all available information (i.e. instance data and ontology schema) and according to the evaluations, it outperforms the traditional personalised and non-personalised methods in terms of accuracy, serendipity and diversity (see Section 5.3.3).

- An evidence on the impact of the resource informativeness on the accuracy, serendipity and diversity of recommendations. We show that resource informativeness has positive impact on the accuracy and negative on the serendipity and diversity of the Linked Data recommendations (see Section 5.3.3).

- An evidence on the trade-off between accuracy, serendipity and diversity of the recommendations. We show that a result set with more serendipitous and more diverse gives less accurate recommendations (see Section 5.3.3).

The work described in this chapter is guided by the following set of research questions on which we provide answers:

- RQ3.1: “How does the quality of our Linked Data recommendation method compare to the traditional personalised and non-personalised methods on a dataset from the Web services domain?” We investigate how our Linked Data recommendation method compares to the other related methods (see Section 5.3.3).

- RQ3.2: “To what extent the resource informativeness influences the accuracy, serendipity and diversity of the Linked Data recommendations?” We investigate the impact of the resource informativeness on the accuracy, serendipity and diversity of the recommendations (see Section 5.3.3).

- RQ3.3: “What is the impact of the serendipity and diversity on the accuracy of the recommendations?” We analyze the trade-off between the serendipity, diversity and the accuracy of recommendations (see Section 5.3.3).

This chapter provides answer also on the following research question, which is related to our work on “knowledge acquisition and semantization”:

- RQ1.2: “To what extent do semantics improve the accuracy of the process of Web service retrieval?” We evaluate evaluate the impact of the semantics on the accuracy of the Web service retrieval. (see Section 5.3.3).

This chapter is based on a EKAW 2014 research paper [A.2] and a poster paper [A.4]. Our poster paper “Personalised, Serendipitous and Diverse Linked Data Resource Recommendations” won best poster award at the EKAW 2014 conference.

1http://www.ida.liu.se/conferences/EKAW14/awards.html
5.1 Introduction

In the past years, the Semantic Web community has been primarily focused on developing technical infrastructure and technologies to make the Web of Data feasible [115]. Consequently, these efforts led the development of various methods for Linked Data acquisition, publishing and interlinking, which gave birth to 1,091 Linked Datasets (as of April 2014 [116]), which is an overall growth of 271% compared to only 294 datasets published in September 2011. Along with these efforts, many end-user applications that consume and deliver Linked Data have been developed. Between the most studied applications which leverage Linked Data are recommender systems. In a nutshell, Linked Data based recommender systems produce recommendations of Linked Data resources representing items of interest. To predict the resources of interest, they exploit the relations and interactions of the users with the resources. The problem of recommendation of Linked Data resources has been addressed in several recent works [22, 30, 117, 31, 19, 20]. However, proposed methods are primarily developed for a specific domain, they require manual pre-processing of the datasets and they can hardly be adapted to new datasets. For example, the method described in [30] uses a multi-domain dataset, but requires to select the subset of information (triples) that will represent the domain. In other words, the properties relevant for the domain of interest (movies) should be manually selected. Similarly, the method described in [23] is also domain-specific, developed for the movie domain, and it requires that a user specify the relevant properties from the specific dataset (i.e. DBpedia). Although, DBpedia is known as multi-domain dataset and the method is using DBpedia, it is still limited to a particular domain (movies), manually defined with a set of domain-specific properties. In summary, there is a need of new sophisticated methods which will be robust enough to process Linked Data from different domains and provide accurate, while at the same time serendipitous and diverse Linked Data resource recommendations.

In this chapter, we present a method for personalised retrieval of Linked Data. The method recommends resources of interest for a user. It relies on the assumption that if person A and person B have interest in similar resources, then person A is likely to have interest in similar resources in the future, as person B. To predict resources of interest, the method first measures the similarity between resources representing users, and then recommends resources from similar users. To measure the similarity between two resources in a Linked Data dataset, we propose a novel similarity measure which primarily relies on the shared information of the resources in an RDF graph. The similarity of the resources we compute based on their shared context (i.e., overlap of the surrounding RDF sub-graphs), which we call resource context graphs. When computing the similarity of the resources, our method takes into consideration 1) the size of shared context—the amount of common resources, 2) the connectivity of each shared resource—how well are the context resources connected with the users’ resources, and 3) the informativeness of each shared resource—less probable resources are considered to be more specific, and consequently more informative than the more common ones. The resources’ informativeness is primarily incorporated to differentiate informative shared resources from non-informative, such as resources of type owl:Thing or skos:Concept. A prototype of the method was implemented on top
5. Personalized Retrieval of Linked Data

Figure 5.1: Excerpt from the Linked Web APIs dataset with resource context graphs with context distance of 3.

of the neo4j\(^2\) graph database and we show a resource recommendation use case in the Web services domain. We evaluate the method on several experiments showing that the method produces highly accurate, serendipitous and diverse recommendations compared to the traditional recommendation techniques. For the evaluation we use a real-world dataset, the Linked Web APIs dataset, which is the main result from the first activity of the thesis (Chapter 3).

5.2 Personalised Resource Recommendation

We formulate the problem of personalised recommendation of resources as problem of ranking and recommending top-N most relevant resources. We base our method on the collaborative filtering technique: it estimates the similarity between users, and produces resource recommendations from users with similar tastes. To this end, we develop two novel graph-based metrics: 1) for measuring semantic resource similarity, and 2) for measuring semantic resource relevance. The first, we use to compute similarity between users represented as RDF resources. The second, uses the computed user similarities to estimate the relevance for each resource candidate.

The metric for measuring semantic resource similarity we develop based on a set of assumptions. The set of assumptions is as follows.

\(^2\url{http://www.neo4j.org/}\)
5.2. Personalised Resource Recommendation

(i) **The more information two resources share, the more similar they are.**
The first assumption is that if two resources share some information, then they are
similar to each other. Considering an RDF graph, a shared information, as described
in [23], might be an object of triples where subjects are the resources in question. In
this case, only shared information in distance of one will be taken into consideration,
when estimating their similarity. However, depending on the way the RDF data is
modelled, similar resources might share information in any distance. Thus, in our
method we allow adjustment of the context distance as required. Figure 5.1 shows an
excerpt of the Linked Web APIs dataset (see Chapter 3 for more information about
the dataset). In the figure, we present two context graphs with a distance of 3, for the
users *Alfredo* and *mlachwani*. Considering the figure, the users have 6 resources in
common. Note that if we choose a lower distance values, 1 or 2, no shared information
will be evidenced.

(ii) **Better connected shared resources carry more similarity information.**
According to the assumption, for the user *Alfredo*, the Twitter-API carries more
similarity information than the Facebook-API or the search tag, since the Twitter-
API is better connected to the resource representing the user *Alfredo*. This can be
evaluated by counting the simple paths\(^3\) with a pre-defined maximum length, between
the resources. From the *Alfredo*'s node the Twitter-API can be reached by two simple
paths

\[ p_1 = \{Alfredo, Hashtagram, Twitter-API\} \]
\[ p_2 = \{Alfredo, FriendLynx, Twitter-API\} \]

, while the Facebook-API only by one
\[ p_3 = \{Alfredo, FriendLynx, Facebook-API\} \]

(iii) **Less probable shared resources carry more similarity information than
the more common.** Our assumption is that if two resources have in common more
informative resources, then they are more similar. Considering the whole Linked
Web APIs dataset, the Microsoft-Bing-API carries more information content, since
its node is characterised with a low degree value 40 (due to its low usage in mashups,
leading to a low number of incident links). On the other hand, the Twitter-API and
Facebook-API are popular Web APIs and extensively used in mashups, and their
node degree values are 799 and 418, respectively. To conclude, the Microsoft-Bing-
API is more informative than the Twitter-API and Facebook-API and will carry
more similarity information.

Based on these assumptions, we develop our method for personalised resources recom-
mendation. First, we propose a theoretical definition of Linked Data, followed by several
\(^3\)Note that by *simple path* we mean a path without repeating vertices, as it is defined in the graph
theory.
5. Personalized Retrieval of Linked Data

definitions that we use to ground our metrics for computation of resource similarity and relevance. We present the algorithm that we use to compute the semantic similarity between resources, and the algorithm that uses the computed resource similarity to recommend relevant resources from similar users.

5.2.1 Definitions

Definition 1. Let \( G \) be a Linked Data dataset defined as a graph \( G = (\mathcal{R}, \mathcal{L}) \) in which \( \mathcal{R} = \{r_1, r_2, ..., r_n\} \) is a set of resources identified with their URIs, and \( \mathcal{L} = \{l_1, l_2, ..., l_m\} \) is a set of links (predicates) between those resources, where \( l_i = (r_j, r_k) \) is a concrete link between two resources.

While this definition describes one dataset, the LOD cloud can be described as union of all \( G_i \) datasets. Note that ontologies are not excluded from the definition, and they can be also modelled.

Definition 2. Let \( G_{r_i,d} = (\mathcal{R}_i, \mathcal{L}_i) \) be a sub-graph of a Linked Data dataset graph \( G \) whose resources \( (\mathcal{R}_i) \) and links \( (\mathcal{L}_i) \) sets are subset of those of \( G \) with restriction that only nodes within maximum distance \( d \) from the resource \( r_i \) are included. We will further refer to this sub-graph as a resource context graph.

Definition 3. Let \( C_{r_i,r_j} \) be a set of resources shared by context graphs of the resources \( r_i \) and \( r_j \). We will refer to this set of resources as a shared context.

According to the assumption (iii), in order to give less impact to the less informative resources, we perform weightening of the resources based on the information content (IC) they convey. In the information theory [118], the information content of a concept is defined as the logarithm (i.e., with base 2) of the inverse of its probability

\[
IC(c) = -\log(\pi(c)),
\]

where \( \pi(c) \) is the probability of occurrence of the concept \( c \). The probability \( \pi(c) \) is calculated as the quotient of the frequency of occurrence of \( c \) and the total number of concepts in the corpus. In the following definition we adopt the general definition of IC to be applicable in Linked Data.

Definition 4. Let \( RIC(r_i) \) be a function which computes the IC carried by a resource (RIC) defined as

\[
RIC(r_i) = -\log\left(\frac{\text{deg}(r_i)}{\max\{\text{deg}(r_k) : r_k \in \mathcal{R}\}}\right)
\]

where the probability of occurrence of a resource is computed as the quotient of \( \text{deg}(r_i) \) – resource degree computed as the total number of incident links, and \( \max\{\text{deg}(r_k) : r_k \in \mathcal{R}\} \) – the degree of the resource with the highest degree. Computed resource information content is within the interval \([0,1]\). See Section 5.3.2 for actual computed information content of the resources in the Linked Web APIs dataset.
Definition 5. Let \( \text{gain}(p) \) be a function which computes the gain of information from one end to another in a simple path \( p = \{r_1, r_2, ..., r_n\} \) where \( r_i \) is the \( i \)-th resource in the list of resources visited in the path from \( r_1 \) to \( r_n \). We define the function for computing the information gain as

\[
\text{gain}(p) = \prod_{i=1}^{n} RIC(r_i)
\]  

(5.3)

Note that the gain function is a multiplicative function of RIC weights with values between 0 and 1, and computed gain for longer paths will be lower than for shorter paths. We use the function to compute the connectivity of a shared resource with a user’s resource. For a closer shared resource (shorter path) the computed gain will be higher, than for the more distant shared resource (longer path).
5. **Personalized Retrieval of Linked Data**

5.2.2 **Algorithm: “computing resource similarity”**

The similarity between two resources $r_i$ and $r_j$ we compute according to the following algorithm.

**Algorithm 1: Computing resource similarity.**

**Inputs:**
- A graph $G$ representing a Linked Data dataset.
- A context graph $G_{r_i,d_i} = (R_i, L_i)$ for $r_i$ with context distance $d_i$.
- A context graph $G_{r_j,d_j} = (R_j, L_j)$ for $r_j$ with context distance $d_j$.
- A shared context set $C = \{c_1, c_2, ..., c_n\}$ of the context graphs $G_{r_i,d_i}$ and $G_{r_j,d_j}$.

**Output:**
- A computed similarity $sim_{ij}$ for the resources $r_i$ and $r_j$.

**Uses:**
- A shared context set $C = \{c_1, c_2, ..., c_n\}$ of the context graphs $G_{r_i,d_i}$ and $G_{r_j,d_j}$.

```plaintext
begin
// compute the amount of similarity between the two resources
// as a sum of the similarity carried by each shared resource in $C$
\hspace{1em} sim_{ij} \leftarrow 0
for all $c_k \in C$ do
  // sum the information gained in all simple paths between
  // the resource $r_i$ and the shared context resource $c_k$
  $P_i \leftarrow \text{paths}(r_i, c_k, d_i)$, $s_{r_i} \leftarrow 0$
  \hspace{1em} $s_{r_i} \leftarrow s_{r_i} + \sum_{p \in P_i} \text{gain}(p)$
  // sum the information gained in all simple paths between
  // the resource $r_j$ and the shared context resource $c_k$
  $P_j \leftarrow \text{paths}(r_j, c_k, d_j)$, $s_{r_j} \leftarrow 0$
  \hspace{1em} $s_{r_j} \leftarrow s_{r_j} + \sum_{p \in P_j} \text{gain}(p)$
  \hspace{1em} $sim_{ij} \leftarrow \frac{sim_{ij} + s_{r_i} + s_{r_j}}{2}$
end for
end
```

For each shared context resource $c_k$, the algorithm first retrieves all simple paths (lines 8 and 12) between the shared context resource $c_k$ and the resources we compute similarity for ($r_i$ and $r_j$). Next, in lines 9 and 13, the algorithm computes the gained information for each simple path taking into account the pre-computed resource informativeness. The algorithm independently computes the semantic similarity of the shared context resource $c_k$ to the both resources ($r_i$ and $r_j$). Finally, in line 14, the algorithm computes the semantic similarity carried by a single context resource, as an arithmetic mean of the computed
similarity to the both resources \((r_i \text{ and } r_j)\). The final similarity score is computed as sum of the similarity information carried by each context resource.

5.2.3 Algorithm: “computing resource relevance”

The computed resource similarity using the previous algorithm, is then used to compute the relevance of a resource candidate for a given user. The relevance of the resource for a user we compute according to the following algorithm.

Algorithm 2: Computing resource relevance.

Inputs:
- Graph \(G\) representing a Linked Data dataset.
- Resources \(r_u\) - a user requester, and \(r_c\) - a resource candidate.
- A set of users’ resources \(R' = \{r_1, r_2, ..., r_n\}\), where \(r_k \in R \setminus r_u\).
- A set of user similarity scores \(S = \{s_{u1}, s_{u2}, ..., s_{un}\}\), where \(s_{uk}\) is a semantic similarity computed with Alg 1 for the resource \(r_u\) and \(r_k \in R'\).

Output:
- A computed relevance score for the resource candidate \(r_c\) and the user \(r_i\).

Uses:
- A function \(C\) that returns a resource context graph for a given resource.
- A function \(paths(r_i, r_j, d)\) that returns a set of all simple paths (with a maximum length \(d\)) between two resources.
- A function \(gain(p)\) that computes the gain of information in a path \(p\).

1 begin
2 \(rel \leftarrow 0\)
3 for all \(r_k \in R'\) do
4     // create a resource context graph
5     \(G_{rk} \leftarrow C(r_k, d, G)\)
6     // check presence of the resource \(r_c\) in the context graph
7     if \(r_c \in G_{rk}\) then
8         // sum the gain of information for all simple paths between
9         // the user and the resource candidate
10        \(P \leftarrow paths(r_c, r_k, d)\)
11        \(rel \leftarrow rel + s_{uk} \star \sum_{p_i \in P} gain(p_i)\)
12     end if
13 end for
14 end
First, the algorithm creates a context graph for each user similar with the user $r_u$ (line 5). Next, the algorithm checks whether the resource candidate is present in the context graph (line 7). If yes, the algorithm computes the connectivity of the similar user and the resource candidate $r_c$ (lines 10–11). The connectivity is computed as sum of the gained information for all the simple paths between the user and the resource candidate $r_c$. In line 11, the algorithm also takes into account the pre-computed similarity score $s_{uk}$ between the users. The final score is a sum of the relevance values computed from each similar user.

5.3 Experimental Evaluation

In this section, we briefly describe the dataset used for validation and evaluation of the method. We present a resource recommendation use case and we report on the results from several experiments. In the experiments we addressed following set of questions:

- What is the quality of the recommendations provided by our method in comparison with the other traditional methods?
- How the resource information content (RIC) influences the quality of the recommendations?
- How surprising and diverse recommendations generates the method?

5.3.1 Dataset Description

In order to validate and evaluate the method, we opted for the Linked Web APIs dataset [A.1]. The Linked Web APIs dataset is the main result from the knowledge acquisition and semantization activity of the thesis, which is described in Chapter 3.

In the experiments we used a snapshot of the Linked Web Dataset as of April 24th, 2014. This particular snapshot contains over 170K RDF triples describing 11,339 APIs, 7,415 mashups and 5,907 users. Figure 5.1 shows an excerpt of the dataset.

5.3.2 Use-case: Resources Recommendations

In order to validate and demonstrate our method, we developed a resource recommendation use case for the Web services domain: recommendation of resources representing Web APIs. For this purpose we used the Linked Web APIs dataset. After loading the dataset, for each resource we compute its information content (see definition 4). Table 5.1 shows the top 5 resources with highest and lowest information content.

As expected, resources which are more distinctive, have higher RIC, and will have higher influence in the similarity computation, while the most probable resources, are less informative, and will have less influence in the similarity computation. For example, the resources representing ontological classes, such as the *wl:Service* or *ls:Mashup* class, carry less RIC due to their high degree value in the RDF graph. On the other side, sharing
5.3. Experimental Evaluation

Table 5.1: Top 5 resources with highest (left) and lowest RIC (right).

<table>
<thead>
<tr>
<th>Resource ID</th>
<th>Label</th>
<th>RIC (bits)</th>
<th>Resource ID</th>
<th>Label</th>
<th>RIC (bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>27766</td>
<td>Paigeadele user</td>
<td>1.00000</td>
<td>7</td>
<td>Service class</td>
<td>0.00000</td>
</tr>
<tr>
<td>27871</td>
<td>retouching tag</td>
<td>0.92576</td>
<td>13</td>
<td>Mashup class</td>
<td>0.04550</td>
</tr>
<tr>
<td>28017</td>
<td>Pbwiki API</td>
<td>0.88233</td>
<td>34</td>
<td>Person class</td>
<td>0.06985</td>
</tr>
<tr>
<td>28015</td>
<td>Usefulbytes API</td>
<td>0.85151</td>
<td>39</td>
<td>Tag class</td>
<td>0.13267</td>
</tr>
<tr>
<td>28014</td>
<td>Philly add API</td>
<td>0.82761</td>
<td>12</td>
<td>mapping tag</td>
<td>0.13273</td>
</tr>
</tbody>
</table>

resource representing the tag *retouching* or the *Usefulbytes API*, will carry more RIC due to their low degree value.

Next, using the Algorithm 1 we compute the similarity between the resources representing users. In the Linked Web APIs dataset those are instances of the *foaf:Person* class. When computing the resource similarity it is necessary to set the context distance of resource context graphs (see definition 2). We experimentally set the context distance to 2, thus only resources within distance of 2 will be taken into account when creating the resource context graphs. In this case, only resources representing mashups, Web APIs and assigned tags will be present in the context graphs. See Section 5.4 for discussion on setting the resource context distance.

Finally, using the Algorithm 2 we compute the relevance between each user and each resource candidate. Here, we focused on computation of relevance only for instances of the class *wl:Service*, however, the relevance can be be computed for any other resources, e.g., categories, mashups and even users. Table 5.2 shows the top 5 most similar users with the user *Alfredo* and the top 5 Web APIs with highest relevance score, also for the user *Alfredo*.

Table 5.2: Top 5 most similar users with the user *Alfredo* (left) and the top 5 most relevant Web APIs (right).

<table>
<thead>
<tr>
<th>Resource ID</th>
<th>Username</th>
<th>Similarity score</th>
<th>Resource ID</th>
<th>API Name</th>
<th>Relevance score</th>
</tr>
</thead>
<tbody>
<tr>
<td>511</td>
<td>Avishai</td>
<td>2.11250</td>
<td>245</td>
<td>Twitter API</td>
<td>49.78257</td>
</tr>
<tr>
<td>731</td>
<td>Frogcologne</td>
<td>1.79806</td>
<td>10</td>
<td>Google Maps API</td>
<td>36.32023</td>
</tr>
<tr>
<td>20130</td>
<td>Nobosh</td>
<td>1.69410</td>
<td>129</td>
<td>Facebook API</td>
<td>33.34930</td>
</tr>
<tr>
<td>2505</td>
<td>Tripsailor</td>
<td>1.64018</td>
<td>331</td>
<td>Box API</td>
<td>27.85667</td>
</tr>
<tr>
<td>1407</td>
<td>Rakfl</td>
<td>1.63710</td>
<td>165</td>
<td>Flickr API</td>
<td>24.60448</td>
</tr>
</tbody>
</table>
5.3.3 Evaluation

In order to evaluate the quality of our method in terms of accuracy and usefulness of the recommendations, we followed standard evaluation protocols for recommender systems. For the evaluation we used the Linked Web APIs dataset and we randomly created training (80%) and testing partition (20%). This led to creation of 3,089 test cases.

For the evaluation of the accuracy we focused on several standard well-known metrics used for evaluation of recommender systems [119]. The metrics used for the evaluation of the accuracy are as follows.

- **Precision and Recall.** A classical evaluation metrics where precision is defined as a fraction of the retrieved items that are relevant, and recall is defined as a fraction of the relevant documents that are retrieved.

- **Area Under the Curve (AUC).** Measures the quality of a list of ranked items. The AUC is equivalent to the probability that the recommender will rank a randomly chosen positive instance higher than a randomly chosen negative instance. For a random recommender it can be expected to get half the positives and half the negatives correct with an AUC value close to 0.5.

- **Normalized Discounted Cumulative Gain (NDCG).** Measures the quality of a list of ranked items taking into account the position of each item. It gives higher weight to items with higher rank.

- **Mean Average Precision (MAP).** Measures the quality of the list of ranked items as mean of the average precision for a set of test queries.

- **Mean Reciprocal Rank (MRR).** Considers the rank position of the items in the ranking list. A reciprocal rank for a single query is computed as a reciprocal of the rank at which the relevant item is retrieved.

5.3.3.1 Experiments Set 1: Evaluation of the Accuracy of the Recommendations

In order to evaluate how the information content influences the accuracy, we evaluated the accuracy on two variants of our method. One which takes into account the informativeness of the resources, and one which does not. For the latter, we experimentally set the informativeness for all the resources at a fixed value of 0.9. Note that choosing any value in the interval between 0 and 1 will have same effect on the final results. We also performed a comparison of our method with the traditional personalised collaborative filtering methods (User-KNN and Item-KNN)\(^4\) and non-personalised methods (Random\(^5\) and Most popular\(^6\)), which we consider as baseline. The evaluation was conducted using

---

\(^4\)For the UserKNN and ItemKNN baseline methods, was used a cosine similarity function with the default neighbourhood size experimentally set to k=80.

\(^5\)Random recommender - randomly recommends items from a given set.

\(^6\)Most popular recommender - recommends items weighted by the number of times they have been seen in the past.
the evaluation environment MyMediaLite\textsuperscript{7} v3.10, which also provides implementation of the evaluated metrics and baseline algorithms. Figure 5.2 shows the Precision and Recall curves obtained for different methods.

![Figure 5.2: Precision and Recall curves obtained for different methods.\textsuperscript{8}](image)

The results show that our method outperforms the traditional personalised User-KNN and Item-KNN recommendation methods, as well as the simple Random and Most popular recommendation methods. The results also show that our method achieves better results when taking into account the resource information content. From all the evaluated methods, the lowest results were achieved for the Random recommender. Slightly better results were achieved by the Item-KNN, followed by the User-KNN. It is interesting the fact that the Most popular method achieved better results compared to the baseline methods. Most likely it is due to the long-tail distribution of the Web API usage in mashups, where small number of Web APIs enjoy significantly greater popularity than the others \[120, 121\].

The results for the AUC, NDCG, MAP and MRR metrics are summarized in Table 5.3. It can be observed that also for the other metrics our method outperforms the baseline methods. Here we can again see that the variant of our method, which takes into account the informativeness of the resources achieves better results over the variant which does not. An improvement of 6.65\% was achieved for AUC, 16.98\% for NDCG, 16.68\% for MRR, and 15.64\% for MRR.

\textsuperscript{7}MyMediaLite evaluation environment - \url{http://mymedialite.net/}

\textsuperscript{8}The precision/recall curves were obtained looking @topN, with N set to \{5;10;15;20;30;40;50;60;70;80;90;100;150;200\}.
Table 5.3: Evaluation results for: Area Under the Curve (AUC), Normalized Discounted Cumulative Gain (NDCG), Mean Average Precision (MAP), Mean Reciprocal Rank (MRR)

<table>
<thead>
<tr>
<th></th>
<th>Random</th>
<th>Most popular</th>
<th>User-KNN</th>
<th>Item-KNN</th>
<th>Linked Data based without RIC</th>
<th>Linked Data based with RIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUC</td>
<td>0.50831</td>
<td>0.89072</td>
<td>0.64023</td>
<td>0.71038</td>
<td>0.89162</td>
<td>0.95093</td>
</tr>
<tr>
<td>NDCG</td>
<td>0.11608</td>
<td>0.38547</td>
<td>0.22278</td>
<td>0.11273</td>
<td>0.59401</td>
<td>0.69486</td>
</tr>
<tr>
<td>MAP</td>
<td>0.0064</td>
<td>0.26235</td>
<td>0.14506</td>
<td>0.02114</td>
<td>0.53442</td>
<td>0.62358</td>
</tr>
<tr>
<td>MRR</td>
<td>0.00742</td>
<td>0.2946</td>
<td>0.17653</td>
<td>0.02355</td>
<td>0.57835</td>
<td>0.66882</td>
</tr>
</tbody>
</table>

5.3.3.2 Experiments Set 2: Evaluation of the Serendipity and Diversity of the Recommendations

Apart from the accuracy, another important dimension of the recommender system, as argued in [119], is the usefulness of the recommendations in terms of “how surprising and diverse the recommendations are”. Since in our case the user requester and the recommended items are represented as nodes in graph, we define the serendipity as the length of the shortest path between the requester’s resource ($r_u$) and the recommended resource ($r_i$). A larger value of the shortest path indicates greater surprise. The overall serendipity of a set of resources (set $C$) is the average serendipity of the resources in the set.

$$\text{Serendipity}(r_u, C) = \frac{\sum_{r_i \in C} \text{shortest-path}(r_u, r_i)}{|C|}$$  \hspace{1cm} (5.4)

The diversity of a set of recommended resources we compute as the average dissimilarity among all resource pairs. The formula used for computation of diversity is as follows.

$$\text{Diversity}(C) = \frac{\sum_{r_i \in C} \sum_{r_j \in C-\{r_i\}} \left(1 - \text{similarity}(r_i, r_j)\right)}{|C|(|C|-1)/2}$$ \hspace{1cm} (5.5)

Here, the similarity between the resources we compute as the Jaccard coefficient of the context graphs of the resources (each $r_i$ and $r_j$) in the set of recommendations $C$. Computed diversity score is in the $[0, 1]$ interval, where values close to 0 indicates very similar set of recommended resource, and close to 1 very diverse resource recommendations. In the Web services recommendation use case, diverse recommendations can be considered those recommendations where the Web APIs belong to different category, have assigned different tags, support different protocols or data formats, or have been used by different users.

We evaluate the serendipity and diversity looking at the top 5, 10, 15 and 20 recommendations. The results from the evaluation of the serendipity and diversity are summarised in Table 5.4. The results from the evaluation of serendipity and diversity show that our method outperforms the other methods. It can be also observed that the variant of our method which does not consider the informativeness of the resources produce more serendipitous and diverse recommendations compared to the variant which considers the...
5.3. Experimental Evaluation

Table 5.4: Results from the evaluation of serendipity and diversity

<table>
<thead>
<tr>
<th>@top-N</th>
<th>Random</th>
<th>Most Popular</th>
<th>User-KNN</th>
<th>Item-KNN</th>
<th>Linked Data based without RIC</th>
<th>Linked Data based with RIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>@top-5</td>
<td>2.97752</td>
<td>2.66810</td>
<td>2.59197</td>
<td>2.68006</td>
<td>3.18881</td>
<td>3.03271</td>
</tr>
<tr>
<td>@top-10</td>
<td>2.98455</td>
<td>2.67465</td>
<td>2.65514</td>
<td>2.70402</td>
<td>3.54821</td>
<td>3.26700</td>
</tr>
<tr>
<td>@top-15</td>
<td>2.98364</td>
<td>2.65816</td>
<td>2.68101</td>
<td>2.71267</td>
<td>3.73117</td>
<td>3.36509</td>
</tr>
<tr>
<td>@top-20</td>
<td>2.98455</td>
<td>2.65184</td>
<td>2.69780</td>
<td>2.70968</td>
<td>3.84142</td>
<td>3.42444</td>
</tr>
</tbody>
</table>

informativeness. We can conclude that there is a trade-off between accuracy and serendipity/diversity which is directly influenced by the resource informativeness. In other words, when considering the resource informativeness our method provides more accurate but less serendipitous and diverse recommendations.

We also studied the optimal trade-off between the precision/recall and serendipity/diversity. Figure 5.3 depicts the obtained trade-off curves for our method.

![Trade-off between serendipity and accuracy](image)

Figure 5.3: Trade-off between serendipity and accuracy studied @top 5, 10, 15 and 20

The results show that the optimal values are: i) precision 0.12, recall 0.77 and serendipity 3.2, ii) precision 0.13, recall 0.763 and diversity 0.825. It can be also observed that the optimal precision/recall and serendipity/diversity is achieved when recommending between the top 5 and top 10 most relevant resources.
5.4 Discussion and Future Work

5.4.1 Comparison to Related Work

A particular method that relates to ours is the dbrec method presented in [30]. The method is supported by a semantic distance measure for measuring relatedness between resources. The measure is defined as a function of the direct and indirect links between resources. One limitation of this measure is that similarity between resources can be measured only if the graph distance between the resources is not more than two. Since the smallest distance between the users in the Linked Web APIs dataset is four (i.e., foaf:Person → ls:Mashup → wl:Service → ls:Mashup → foaf:Person), the measure will fail to produce recommendations for the Linked Web APIs dataset. In comparison, our method can be easily adapted to any dataset by setting the resource context distance parameter (see Section 5.4.2 for a discussion on the context distance parameter). Moreover, although the DBrec method has been validated on different domains found in DBpedia, it still demands manual pre-processing of the dataset. Also, in spite of the fact that it is using a multi-domain dataset, the relevant properties for the targeted domain (movies) have to be specified. In contrast, our method is not domain or dataset specific and does not require any manual pre-processing of the datasets – it exploits RDF datasets in their original form.

In [23] the authors propose a Linked Data enabled content-based movie recommender, which uses a vector space model to compute similarities between the movies. However, the approach is not suitable for computation of similarities of resources in datasets, such as the Linked Web APIs dataset, where the graph distance between the resources is more than two. Moreover, the approach has been exclusively developed for the movies domain [22] and its adaptation to other domains requires manual pre-processing; the relevant properties have to be specified. In comparison, as already discussed, our method does not require manual pre-processing of the dataset. Another method, which uses the vector space model to compute similarities between entities is the one described in [117]. However, since similarities only between directly linked resources can be computed, this method is also not applicable on datasets such as the Linked Web APIs dataset. The problem with dataset adaptation has been addressed by the RecSPARQL method [26], which enables development of customized recommender for an arbitrary RDF graph. However, RecSPARQL requires human expertise to configure the method and select the features to be used when recommending resources of interest. In our method, every piece of information is used as feature, while the actually weighting of the features is done by computation of their informativeness.

Lookup Explore Discovery (LED) [31] and Discovery Hub [19] are two exploratory search engines which utilize DBpedia and recommend DBpedia resources. However, they explore only small portion of the available information in DBpedia. LED exploits only the abstracts, labels and the page links, while DiscoveryHub relies only on page links and information encoded as triples with properties dcterms:subject and rdf:type. In comparison, our method considers every piece of information (triples) of the given dataset.

Furthermore, both, instance data and the ontology schema, can provide valuable in-
formation for the Linked Data Recommenders. However, vast majority of Linked Data recommenders only consider the instance data, while the schema has been only considered in [31, 23, 24, 26]. In comparison, in our method we consider both, the ontology schema and the instance data.

With regards to the resource informativeness, the only existing approach which considers the resource informativeness can be found in [122]. The informativeness of the resources is computed as sum of the information content of its features (directly linked resources). Thus, a resource linked to another resource with high information content, will result also in a high informative resource. In comparison, in our approach we compute the informativeness of the resources based on the number of in-out links incident with the resource.

For a complete review of the existing Linked Data recommenders we refer the reader to Section 2.2.3.

5.4.2 Setting the Resource Context Distance

The resource context distance allows us to control the amount of context used when computing the resource similarity. The larger context distance we set, the more context is considered. For example, with distance set to 1, only directly linked resources will be used as context. In datasets, where users in an RDF graph are close to each other, will require setting lower distance, while in datasets, where users are far, will require higher distance. Choosing small context distance in datasets where the users are far from each other, can lead to possibly no overlap of the resource context graphs, and consequently no similarity computed. In our experiments, we set the distance to two, and thus, the context of the user resources will contain resources representing the mashups the user created, the Web APIs used in the mashups and the assigned tags. Also, it is obvious that the size of the context directly influences the time required for computation of the resource similarity. In our future work, we would like to explore methods for automatic determination of optimal context distance for a given dataset.

5.4.3 Resource Similarity Computation in Multi-Domain Datasets

When computing resource similarity our method uses the shared resource context. While the Linked Web APIs is a single-domain dataset, in a multi-domain datasets, such as DBpedia, the shared contexts might contain resources from various domains, which might have direct influence on the recommendations. For illustration, two users being similar in the music domain, does not mean they are similar also in the Web service domain. In our future work, we would like to explore such situations, assess their impact on the quality of the recommendations and appropriately adapt our method. Last but not least we want to evaluate the method on other benchmark datasets with different characteristics and from other domains. This includes, for example, the MovieLens, the DBLP dataset, and the ACM DL dataset.
5.5 Summary

A growing number of published datasets in the LOD cloud require new methods that can provide more efficient retrieval of Linked Data. In this chapter, we have presented a novel configurable method for personalised retrieval of Linked Data. The method can be easily adapted to a dataset from any domain and make use of it. It relies on the collaborative filtering approach and it recommends resources from users with similar resource interests. The method is supported with two novel metrics for computing resource similarity and relevance. When computing the similarity between the users the method primarily takes into account the commonalities, the informativeness and the connectiviteness of the shared resources. We validated the method on a resource recommendation use case from the Web services domain and we presented its capabilities. We also evaluated the method on a real-world dataset and the results show that the method outperforms the traditional personalised collaborative filtering and non-personalised methods in terms of accuracy, serendipity and diversity. The results also show that considering the informativeness of the resources improves the accuracy of the recommendations.

This chapter described our work within the personalized knowledge retrieval activity of the thesis. The work described in this chapter exploits the Linked Web APIs dataset, which is the main result from the two previous activities, the knowledge acquisition and semantization (Chapter 3) and the knowledge extraction and integration activity (Chapter 4).
Conclusions and Future Work

“Knowledge itself is power.”
(Latin: Ipsa Scientia Potestas Est)
– Sir Francis Bacon [123]

This chapter provides a summary of the thesis, revisits the research questions which guided our work and lists the main contributions of the thesis. It also describes the limitations and our plans for future work.

6.1 Summary

Linked Data has become one of the most successful movements endorsed by the Semantic Web community. It became de facto standard for publishing, sharing and integration of data, information and knowledge. Despite the increased popularity of the Linked Data, there are several open challenges that should be addressed. First, although vast amount of knowledge has been published as Linked Data, there are specific domains which lack coverage in the Linked Open Data cloud (e.g. Web services domain). This requires, data acquisition, semantization and publishing of the domain-specific knowledge as part of the LOD cloud. Second, while the ultimate goal of Linked Data is to publish structured information, huge amount of knowledge is still hidden in an unstructured format, in text documents. In order to realize the vision of the Semantic Web, there is need to extract and transform this hidden information into a structured knowledge, and integrate it into the Linked Data space. Integrating this valuable knowledge with the LOD cloud will bring together two data spaces: the Web of Documents and the Web of Data. And third, due to the huge and diverse amount of information, the retrieval of information from the LOD cloud demands significant amount of effort. This requires new mechanisms to address the information overload problem and provide efficient retrieval of Linked Data.
6. Conclusions and Future Work

In this thesis, we presented an overarching process on “Linked Data based Knowledge Provisioning” which addresses the above-mentioned problems. The process spans three activities: i) knowledge acquisition and semantization for the Web services domain, ii) knowledge extraction and integration using salient named entities, and iii) personalized knowledge retrieval in the context of Linked Data. The knowledge acquisition and semantization activity contributes with acquisition of knowledge for the Web services domain. The main results from this activity is a dataset with semantic Web service descriptions, which is the largest dataset of its kind. The dataset is supported with a light-weight ontology for modeling relevant Web service information. The knowledge extraction and integration activity contributes with named entity based methods for knowledge extraction and integration with the LOD cloud (i.e. DBpedia). We developed methods for identification, classification and linking of salient named entities. We also developed an NER system, named Entityclassifier.eu, which integrates the developed methods. The system has been used to link particular information from the Web services dataset with information from the LOD cloud. Finally, the personalized knowledge retrieval activity contributes with a method for efficient retrieval of Linked Data. The main results from this activity is a method for recommendation of Linked Data resources of interest for the user. The method takes into account the commonalities, the informativeness and the connectiviteness of the resources, it is not domain or dataset specific, it does not require pre-processing of the dataset, and as it was shown in a set of experiments, it outperforms the other traditional personalized and non-personalized methods in terms of accuracy, diversity and serendipity. The method was validated on a dataset which is the main contribution from the first activity of the thesis.

6.2 Research Questions and Contributions of the Thesis

In this section, we remind and discuss the defined research questions (RQ) and hypotheses (H) which guided our work. We also summarize the main contributions of the thesis (CT).

6.2.1 Acquisition and Semantization of Web API Descriptions

Due to the lack of a dataset with semantic Web service descriptions published as part of the LOD cloud, in our work we focused on acquisition, semantization and provisioning of knowledge for the Web services domain. As a result, we have created a dataset with semantic Web service descriptions, which is the largest dataset of its kind. Our work on knowledge acquisition and semantization for the Web services domain was guided by the following set of research questions and hypotheses.

- RQ1.1: “What are the benefits of a dataset with semantic Web service descriptions?”
- H1.1: “A dataset with semantic Web Service descriptions enables advanced analysis of the service ecosystem that was not possible before.”
In order to provide an answer on RQ1.1, we have executed a survey and developed several use cases which provide evidence for H1.1 on the capabilities of the dataset. In particular, on several use cases we show that the dataset enables analysis of the ecosystems, such as comparison of the popularity and consumption of different Web services over time, identification of the latest API trends (e.g., protocol or data formats), or analysis of the API’s popularity across different domains (Section 3.7.1). Moreover, we have executed a survey (Section 3.7.2) in which the participants highlighted the main benefits of a dataset with semantic Web service descriptions. According to the survey, 86% of the participants find the dataset useful in searching and selection of relevant APIs, 86% stated that such dataset will increase the visibility of the APIs, 62% agreed that the dataset will enable analysis of the recent trends in the API ecosystem, 59% stated that it enables comparison of APIs, 52% that the dataset can be employed for automated composition of Web APIs, and 38% stated that the dataset can be used to track the popularity of the Web APIs over time. In summary, such analysis of the service ecosystem cannot be executed without the availability of a dataset with semantic Web descriptions.

○ RQ1.2: “To what extent do semantics improve the accuracy of the process of Web service retrieval?”

○ H1.2 “A dataset with semantic Web service descriptions enables more accurate retrieval of Web services.”

We were also interested in how the semantics improve the accuracy of the Web service recommendations. In order to provide an answer on RQ1.2, we developed a method which exploits the semantics and provides personalized retrieval of Linked Data resources (Chapter 5). We applied the method on the dataset with semantic Web service descriptions and executed an experiment where we compared our method with the other traditional non-semantic based recommendation mechanisms (Section 5.3.3, Experiment 1). According to the results from the experiment, our method which considers semantics provides more accurate results compared to the traditional non-semantic based recommendation mechanisms. The results show an increase in the accuracy (according to the Area Under the Curve measure) of 87%, 7%, 49% and 34% for four different non-semantic based recommendation mechanisms. The results from the experiment confirm our hypothesis (H1.2) that a dataset with semantic Web service descriptions enables more accurate retrieval of Web services.

○ RQ1.3: “How can we improve the efficiency of modeling relevant Web service information?”

○ H1.3: “An integrated light-weight ontology enables to efficiently capture all relevant Web service information that was not possible before.”

Our work on RQ1.3 provided evidence on the capabilities of the existing semantic Web service models. According to the literature review, existing semantic Web service models are complex; they do not completely capture the available Web service information or they
6. Conclusions and Future Work

address Web service architectural models which are nowadays not prevalent on the Web. According to our research, none of the existing semantic Web service models can capture all aspects of the available information (i.e. provenance, functional, non-functional, technical and temporal information).

In order to address these issues, we have developed a light-weight ontology which builds on top of the existing ontologies and appropriately extends them (Section 3.3). On a real-world data from the Web services domain (Section 3.4) we show that the ontology can efficiently capture all available Web services information.

○ RQ1.4: “To what extent do different types of Web service users find a dataset with semantic Web service descriptions useful?”

○ H1.4: “Vast majority of the Web service users find a dataset with semantic Web descriptions useful.”

In order to provide answer on RQ1.4, we have executed a survey and evaluated the potential and the usefulness of the dataset as seen by the Web service consumers and providers (Section 3.7.2). According to the results, 17% of the consumers find the dataset very useful, 48% useful and 35% somewhat useful. From the Web service providers, 5% find the dataset very useful, 48% useful, 26% somewhat useful, 16% little useful, and only 5% not useful at all. In overall, the results from the survey show that vast majority of the Web service users find a dataset with semantic Web descriptions useful.

In summary, the main contributions of the thesis (CT) related to Acquisition and Semantization of Web API Descriptions are threefold:

○ CT1.1: A Linked Data dataset with semantic Web API descriptions, which is largest of its kind.

○ CT1.2: A light-weight ontology for modelling relevant Web API information.

○ CT1.3: A survey on the usefulness of the dataset which ascertains the added value and the degree of achievement.

The results from our work on knowledge acquisition, semantization and Linked Data publishing for the Web services domain have been published in a Semantic Web Journal paper [A.1] and an ISWC 2012 research paper [A.3] which was nominated for best research and best student paper award1.

6.2.2 Knowledge Extraction and Integration with Salient Linked Entities

The popularity of the NER and EL solutions have increased over the last few years. In spite of this increase, several problems are starting to surface. Existing solutions for entity recognition and classification are dependent on training data and they do not assess the

1http://iswc2012.semanticweb.org/awards.html
6.2. Research Questions and Contributions of the Thesis

importance of the entities within the document. Moreover, there is lack of a language resources for training entity salience which are complete, publicly available and evaluated by human. In our work on knowledge extraction and integration with salient named entities, we address above-mentioned problems and develop an NER system supported with several methods for identification of salient entities, entity classification and entity linking. Our work on knowledge extraction and integration was guided by the following set of research questions and hypotheses.

- **RQ2.1:** “How does the quality of our NER system compare to other systems for different datasets and for different type of focus queries?”
- **H2.1:** “Our NER system gives more accurate results compared to other related systems.”

In order to provide answer on RQ2.1, we have executed several experimental evaluations and compared our system with several state of the art systems. We also evaluated the performance of our system and methods on different datasets and for different type of focus queries. On the Czech Traveler dataset (Section 4.1.5.1) our system outperformed three state of the art systems and achieved 0.61 F1 for entity spotting (second best by AlchemyAPI 0.58 F1), and 0.66 F1 for entity classification (second best by OpenCalais and AlchemyAPI 0.45 F1). Also, we show that our most-frequent-sense based linking approach (0.67 F1) outperforms the context based linking approach, which is used by DBpedia Spotlight (0.43 F1). Note that our methods for entity spotting and classification are fully unsupervised in comparison to supervised methods implemented as part of the other NER systems. Moreover, the entity type mining task is executed in query time. We also evaluated the performance of our system for different focus queries (PER, ORG, GPE) and different document collections (newswire, web documents, discussion fora documents) at the TAC 2013 challenge (Section 4.1.5.3) and the results show that our most-frequent-sense based linking approach achieved best results for the GPE focus queries (0.677 F1) and for the discussion fora documents (0.539 F1).

- **RQ2.2:** “How accurate results gives our method for identification of salient named entities compared to other similar methods?”
- **H2.2:** “Our method for learning entity salience gives more precise results than the state of the art method propose by Dunietz and Gillick [1].”

In our work on RQ2.2, we have executed an experiment and evaluated our method for learning entity salience. In the experiment, we compared the performance of our entity salience method in terms of accuracy with the most related state of the art method described in [1]. According to the results from the experiment, our method gives more precise results (0.611 F1) compared to Dunietz and Gillick [1] method (0.605 F1) (Section 4.2.4, Experiment 2). The main difference between our method and the other method [1] is in the computation of the global features. Our method considers the complete external entity knowledge graph (i.e. DBpedia), while the method described in [1] considers only a graph
6. Conclusions and Future Work

consisting of entities which occur in the document. Moreover, our method relies on several graph metrics (e.g. PageRank, HITS, in-degree and out-degree), while the other method only on PageRank.

- RQ2.3: “What is the impact of individual and combined local and global set of features on the performance of learning entity salience?”
- H2.3: “A combined set of local and global features gives better accuracy than each set used individually.”

In order to provide answer on RQ2.3, we run an experiment and analyzed the impact of the local and global features on the performance of learning entity salience (Section 4.2.4, Experiment 3). We evaluated the accuracy of a model which considers both set of features (i.e. local and global features), and a model which considers each feature set individually. The results from the experiments confirm our hypothesis (H2.3) that a model which relies on the local and global features gives better results (0.607 F1) than a model which considers each feature set individually (local 0.592 F1; global 0.489 F1).

- RQ2.4: “How does the quality of the entity links influence the performance of learning entity salience?”
- H2.4: “Incorrect links have low impact on the performance of learning entity salience.”

In our work on RQ2.4, we analyzed the impact of the quality of entity linking on the performance of learning entity salience (Section 4.2.4, Experiment 3). The results show that 10% of incorrectly linked entities results in 2.47% decrease of accuracy, while 20-25% of incorrectly linked entities results in 4.12% decrease of the accuracy. It can be also observed, that also with 50% of incorrectly linked entities, the learning accuracy still shows promising results (0.553 F1). According to the results, the quality of the entity links has low impact on the performance of learning entity salience.

In summary, the main contributions of the thesis (CT) related to Knowledge Extraction and Integration with Salient Named Entities are sixfold:

- CT2.1: An open-source NER system named Entityclassifier.eu and a set of supporting methods for entity recognition, linking and classification (Section 4.1). The system is supported with unsupervised methods for entity spotting and classification and several most-frequent-sense based and context-based entity linking methods. In an experiment on the Czech Traveler dataset (Section 4.1.5.1), the system outperformed three other state of the art systems in terms of accuracy.

- CT2.2: An evidence on the accuracy of each developed NER/EL method under different conditions (datasets, domains, focus queries and languages). We show that the most-frequent-sense approach gives more accurate results for Person focused queries then for Organization and Geopolitical queries. We also evaluate the performance for different document collections and show that our system gives best results for
6.2. Research Questions and Contributions of the Thesis

discussion fora documents than for newswire or web documents. These experiments have been executed as part of our participation at the TAC 2013 (Section 4.1.5.3) and TAC 2014 (Section 4.1.5.4) challenges.

- **CT2.3:** A method for learning entity salience based on local and global set of features, which gives more precise results when compared to the related state of the art method [1].
- **CT2.4:** A crowdsourced dataset with entity salience annotations, which is the first complete, publicly available and evaluated by human dataset (Section 4.2.3).
- **CT2.5:** An evidence about the impact of the features on the performance of learning entity salience. We show that a combined set of local and global features gives better accuracy than each set used individually (Section 4.2.4, Experiment 3).
- **CT2.6:** An evidence about the impact of the incorrect links on the performance of learning entity salience. We show that incorrect links have low impact on the performance of learning entity salience (Section 4.2.4, Experiment 4).

Moreover, in our work on knowledge extraction and integration with named entities we have provided also following set of auxiliary contributions:

- Ground-truth datasets for evaluation of NER systems (Section 4.3.2).
- An evaluation framework for NER systems (Section 4.3).

The developed methods and resources, and the results from their evaluation, are described in Chapter 4 and published in several papers [A.5, A.6, A.7, A.8, A.9, A.10], book chapters [A.11, A.12] and technical reports [A.13, A.14, A.15].

6.2.3 Personalized Retrieval of Linked Data

The increased number of Linked Data datasets also increased the complexity of retrieval of relevant information from the LOD cloud. In order to overcome this problem, in the last few years, several Linked Data recommendation methods have been developed. However, these methods have been primarily developed for a particular domain or dataset, the require manual pre-processing of the dataset and exploit only small portion of the available information. In our work on personalized knowledge retrieval, we address the above-mentioned problems and developed a method for personalized retrieval of Linked Data. Our work on personalized knowledge retrieval in the context of Linked Data was guided by the following set of research questions and hypotheses.

- **RQ3.1:** “How does the quality of our Linked Data recommendation method compare to the traditional personalised and non-personalised methods on a dataset from the Web services domain?”
6. Conclusions and Future Work

- H3.1: “Our method, which takes into account the commonalities, the informativeness and the connectiviteness of the resources, outperforms the other personalized and non-personalized methods in terms accuracy, diversity and serendipity.”

In order to provide an answer on RQ3.1, we have executed an experiment and compared our method with several other personalized and non-personalized methods (Section 5.3.3). In the experiment, we used the dataset with semantic Web API descriptions (Section 3.4). The results from the experiments confirm our hypothesis (H3.1) that our method, which considers the dataset semantics, the commonalities, informativeness and the connectiviteness of the resources, outperforms the other personalized and non-personalized methods in terms of accuracy (0.95093 F1 - AUC measure), serendipity (3.42444) and diversity (0.83114).

- RQ3.2: “To what extent the resource informativeness influences the accuracy, serendipity and diversity of the Linked Data recommendations?”
- H3.2: “Resource informativeness improves the accuracy but decreases the serendipity and diversity of Linked Data recommendations.”

In our work on RQ3.2, we have executed an experiment and evaluated the impact of the resource informativeness on the accuracy, serendipity and diversity of the recommendations (Section 5.3.3, Experiment 2). The results confirm our hypothesis (H3.2) that the resource informativeness has a positive impact on the accuracy (16.98% for the AUC measure), and negative impact on the diversity (-10.85%) and serendipity (-6.02%).

- RQ3.3: “What is the impact of the serendipity and diversity on the accuracy of the recommendations?”
- H3.3: “A result set with more serendipitous and more diverse gives less accurate recommendations.”

In order to provide an answer on RQ3.3, we have executed an experiment and evaluated the trade-off between serendipity, diversity and accuracy of the recommendations (Section 5.3.3). The results from the experiment confirm our hypothesis (H3.3) that serendipity and diversity have negative impact on the accuracy of the recommendations. According to the results from the experiment, the optimal values are: for a serendipity at 3.2 is precision at 0.12 and recall at 0.77; and for diversity at 0.825 is precision at 0.13 and recall at 0.763. The results also show that the optimal precision/recall and serendipity/diversity is achieved when recommending between the top 5 and top 10 most relevant resources.

In summary, the main contributions of the thesis (CT) related to Personalized Retrieval of Linked Data are threefold:

- CT3.1: A method for personalized retrieval of Linked Data, which takes into account the commonalities, the informativeness and the connectiviteness of the resources. The method (Section 5.2) can be adapted to any dataset and domain, it does not require
pre-processing of the datasets, it exploits all available information (i.e. instance data and ontology schema) and according to the evaluations, it outperforms the traditional personalised and non-personalised methods in terms of accuracy, serendipity and diversity (Section 5.3.3).

- **CT3.2**: An evidence about the impact of the resource informativeness on the accuracy, serendipity and diversity of recommendations. We show that resource informativeness improves the accuracy but decreases the serendipity and diversity of Linked Data recommendations (Section 5.3.3).

- **CT3.3**: An evidence about the trade-off between accuracy, serendipity and diversity of the recommendations. We show that a result set with more serendipitous and more diverse gives less accurate recommendations (see Section 5.3.3).

The developed method and associated evaluations are described in Chapter 5. The method for personalized retrieval of Linked Data is documented and published in two papers [A.2, A.4]. Our poster paper “Personalised, Serendipitous and Diverse Linked Data Resource Recommendations”\(^2\) won best poster award at the EKAW conference\(^2\).

### 6.3 Future Work

In this section, we describe the limitations and future work with regards to the contributions of the thesis.

First and foremost, in our future work we plan to **evaluate the overall knowledge provisioning process** and investigate how individual activities influence the performance of the succeeding activity. In particular, we plan to evaluate how the quality of the knowledge acquisition and semantization influences the Linked Data recommendations, and how the quality of identification of salient linked entities influences the Linked Data recommendations.

#### 6.3.1 Acquisition and Semantization of Web API Descriptions

In our work on knowledge acquisition and semantization for the Web services domain, we developed the largest Linked Data dataset with semantic Web API descriptions. With respect to the created dataset and the ontology, we have identified the following areas for improvement and future work:

- **Semi-automatic API profiling and metadata maintenance**: currently, the provisioning of the API descriptions is primarily a manual task which requires significant effort. The API descriptions are usually maintained by the API providers or the API repository owners. However, with the increasing number of APIs the maintenance of the API descriptions becomes an expensive and difficult task. As part of our future work...

\(^2\)[http://www.ida.liu.se/conferences/EKAW14/awards.html]
6. Conclusions and Future Work

we plan to explore novel ways for semi-automatic creation of API description profiles and keeping the metadata up-to-date. In particular, we plan to investigate existing code (i.e. code samples) which execute particular API functionalities or apply customized NLP techniques on top of existing text/HTML-based API descriptions with the goal to extract API metadata information and update existing or generate new API description profiles.

- **Dataset enrichment and fusion:** currently, the Linked Web APIs dataset is populated with data from the ProgrammableWeb.com repository. Nevertheless, our ultimate goal is to establish the Linked Web APIs as a central Linked Data hub for Web API descriptions. In order to achieve this goal, we are currently working on enriching the dataset with API descriptions from other data sources. Integration of several additional Web API repositories, such as APIs.io³, APIs guru⁴, API For That⁵ and Exicon⁶, is currently an ongoing effort. See Section 2.2.1.2 for more on existing Web API repositories. This, however, requires methods for fusion of the information derived from different data sources, and alignment of the API description formats, such as APIs.json⁷ and OpenAPI⁸ (previously known as Swagger).

- **Annotation of existing API descriptions:** the HTML based API descriptions provide plain text information. In our future work, we plan to annotate the plain text information with structured information using microdata⁹. This will enable search engines and web crawlers to extract and process the Web API description pages and retrieve structured information. For this purpose, we would need to develop annotation mechanisms for embedding structured information for the Web services domain.

- **Exploration of additional use cases:** the Linked Web APIs dataset has already been used in several works related to personalised recommendation of Web APIs [A.3], personalized retrieval of Linked Data [A.2] and temporal link discovery [A.24]. Last but not least, in our future work we want to explore other applications using the dataset and assess its potential.

6.3.2 Knowledge Extraction and Integration with Salient Linked Entities

In our work on knowledge extraction and integration using salient named entities, we have developed a set of methods for identification, classification and linking of salient named entities. We see following limitations and future work directions:

³http://apis.io/
⁴https://apis.guru/openapi-directory/
⁵http://www.apiforthat.com/about
⁶https://app.exiconglobal.com/api-dir/
⁷http://apisjson.org/index.html
⁸https://www.openapis.org/
⁹https://www.w3.org/TR/microdata/
6.3. Future Work

- **Domain specific optimization:** in our work, we have treated named entity recognition as general domain-independent method. According to the results from the experiments, we have identified that in certain scenarios and datasets different methods perform better than others. For example, in the TAC 2014 challenge we have identified that the most-frequent-sense approach performs better than the context-based, however, for rare entities, the context based gives better results. In our future work, we want to work on a domain specific optimization of NER, based on the given dataset and domain.

- **Learning entity salience:** our method for learning entity salience is based on a supervised machine learning approach which relies on a pre-defined set of features. In our future work, we want to extend the feature set with additional information about entities from public knowledge bases such as DBpedia and Wikidata. Furthermore, although the results from the experiments show promising results, we believe that a large scale dataset can further improve the performance. In our future work, we plan to work on exploitation of public large scale language resources, such as DBpedia Abstracts [A.18], and develop a heuristics based method for learning entity salience. We also plan to adapt and apply our entity salience model on different types of texts such as microposts, video subtitles and music lyrics.

- **User loop feedback:** currently, user feedback has not been considered. User feedback can provide valuable information which can be used to optimize the NER method for the target dataset and domain. In our future work, we plan to work on a loop mechanism for capturing user feedback and methods for NER optimization based on the provided feedback.

### 6.3.3 Personalized Retrieval of Linked Data

In our work on personalized knowledge retrieval in the context of Linked Data, we have developed a method for personalized retrieval of Linked Data resources of interest for the users. With respect to the developed method, we have identified following limitations and future work directions:

- **Dataset adaptation by setting the resource context distance:** the resource context distance allows us to control the amount of context used when computing the resource similarity. The larger context distance we set, the more context is considered. Choosing small context distance in datasets where the users are far from each other, can lead to possibly no overlap of the resource context graphs, and consequently no similarity computed. Also, it is obvious that the size of the context directly influences the time required for computation of the resource similarity. In our future work, we would like to explore methods for an automatic determination of an optimal context distance for a given dataset. We also plan to experiment with datasets from different domains and with different characteristics, such as the MovieLens, the DBLP dataset, and the ACM DL dataset.
6. Conclusions and Future Work

- *Multi-domain adaptation:* While the Linked Web APIs is a single-domain dataset, in multi-domain datasets, such as DBpedia, the shared contexts might contain resources from various domains, which might have direct influence on the recommendations. For illustration, two users being similar in the music domain, does not mean they are similar also in the Web service domain. In our future work, we would like to explore such situations, assess their impact on the quality of the recommendations and appropriately adapt our method.

- *User feedback:* Currently, explicit user feedback is not considered. We assume that the user feedback is already modelled and present as part of the RDF graph. In our future work, we would like to extend our method and provide support for a positive and negative user feedback. The feedback can provide valuable information on the preferences of the user and provide more precise, serendipitous and diverse resource recommendations according to the user preferences.
It has been several years since I applied for a PhD and finally I came to a point, when I realize that the time spend on the PhD was very interesting, joyful and challenging but sometimes also frustrating and disappointing. I am proud of the results I have achieved and I believe other future works will re-use and build on top of my results, or at least benefit from the provided “lessons learned” and develop more advanced technologies for a better future of the human being.

At the beginning it was hard for me to understand why I am working on particular problems, but looking back, I think I was doing what seemed to be sympathetic and practical to me. And I think it was the right choice. While the initial plans were a bit different, still the ultimate goal to "enable more efficient knowledge provisioning" remained unchanged until the end of the PhD. Over the time I managed to identify the domain, technologies and methods that I favor. Initially, the main focus of my research was on the domain of Web services, which over the time extended and comprised Linked Data and Information Extraction technologies. It did not happen all at once, but over the time, by joining new projects (LinkedTV, LOD2 and FREME), and communicating to people from different communities (e.g. DBpedia) managed to find myself doing the things that I like and still believe in.

Sometimes I felt disappointed and frustrated when some paper was rejected, the results from the experiments were not as expected, I did not have the people for a proper discussion on the topic, or the people, which I was expecting a contribution from did not provide or provided only a minor input. However, I always managed to find a reason and motivate myself to push the work further since I knew that was important for me and I must bring it to the end in a form of an accepted paper.

I also learned that PhD is more than just reading papers, implementing conceptual ideas and summarizing the results in papers. It is also about communication and collaboration with people working on similar topics, traveling to events and presenting your work to the community, transfer of knowledge and giving lectures and tutorials, guiding and supervision of other students, reviewing papers, organizing events, and finally, writing new project proposals and acquisition of funding for your follow-up research. I am really happy that
6. Conclusions and Future Work

during the PhD I had the chance to attend events at many nice and exotic places, try local food and drinks, but also meet very nice people from all around the world with different cultural backgrounds and overcome any existing social barriers.

During the PhD I acquired valuable research expertise and skills which I would like to employ in the future and address several interesting and challenging research problems the Semantic Web, NLP and Web Engineering communities are facing. The end of this journey is just a beginning of another exciting journey I am looking forward to!

Milan
Prague
November 15th, 2017
Reviewed Publications of the Author Relevant to the Thesis

Awards and Notable Mentions


Journal Papers

17, 55, 140, 152

Conference Papers

17, 63, 70, 132, 157, 158

The paper has been cited in:
Reviewed Publications of the Author Relevant to the Thesis


17, 55, 63, 70, 152, 158, 163

The paper has been cited in:


Reviewed Publications of the Author Relevant to the Thesis


Reviewed Publications of the Author Relevant to the Thesis


The paper has been cited in:


**Book Chapters**


The book chapter has been cited in:

The author has contributed to the following list of technical reports where the work is directly associated with the thesis.

18, 73, 79, 86, 123, 155

18, 73, 79, 82, 96, 109, 155

18, 73, 79, 102, 104, 155
Other Publications of the Author

During the thesis the author contributed to several other publications as main author or co-author. Although they are not directly associated with the thesis, they indirectly helped the author to increase the quality of the thesis. A list of all other publications\(^\text{10}\) of the author is provided below.


The paper has been cited in:

\(^{10}\)http://www.dojchinovski.mk/research/publications/
Other Publications of the Author


The paper has been cited in:


The paper has been cited in:


Appendix A

Appendix

A.1 Survey Document: Linked Web APIs Dataset

In this section we present the survey document associated on the usefulness of the Linked Web APIs dataset (see Section 3.7.2). The contents of the survey document that was sent out is same as the text bellow. The survey was implemented using the Google Forms survey tool and the results from the survey are also available online\(^1\).

Page 1/6: Introduction

The Linked Web APIs dataset (http://linked-web-apis.fit.cvut.cz/) is a Linked Data dataset with semantic descriptions about Web APIs. It contains over 11,339 of Web APIs descriptions, 7,415 mashups and almost 7,717 mashup developers’ profiles. The Linked Web APIs dataset enables

1. API consumers to **discover relevant APIs**,
2. API providers to increase the **visibility and track the popularity** of their APIs, and
3. API analysts to get better **insight into recent trends** in the Web API ecosystem.

The Linked Web APIs dataset is aiming at publishing Web API descriptions and related information in a machine-readable format, as Linked Data descriptions.

With this survey we aim to learn how individuals and organizations could benefit from this dataset. The survey targets (1) developers consuming and developing APIs, (2) API providers, and (3) software analysts in analyzing recent trends. The results of this survey will be analyzed, documented and made public for the community.

\(^1\)https://dx.doi.org/10.6084/m9.figshare.3459044.v2
A. Appendix

All questions are optional and there is no need to provide information you are not allowed to disclose.

For more on the Linked Web APIs dataset

- check the main landing page (http://linked-web-apis.fit.cvut.cz/)
- query our SPARQL endpoint (http://linked-web-apis.fit.cvut.cz/sparql)
- check an example of an API description (http://linked-web-apis.fit.cvut.cz/resource/google-maps_api), or

Page 2/6: Web API Consumer?

Have you ever searched or used an API?

- Yes (goto page 3)
- No (goto page 4)

Page 3/6: Web API Consumer

I search for APIs by...

- running Google search or using other search engine
- running keyword-based search in service directories such as http://www.programmableweb.com/
- asking other developers for an advice
- Other:

How difficult is to find relevant API?

- Very Hard
- Hard
- Somewhat hard
- Moderate
- Easy

How helpful would it be if you could search for an API using an expressive query language such as SPARQL?
How helpful would it be to find relevant API if there was a central API repository that you could query instead of searching for API at different locations?

- Very helpful
- Helpful
- Somewhat helpful
- Little helpful
- Not helpful

*How useful do you find the Linked Web APIs dataset from perspective of Web API consumer?* E.g. search for relevant APIs, browse the dataset and learn from existing API mashups, mine recent trends in protocols/formats, etc.

- Very useful
- Useful
- Somewhat useful
- Little useful
- Not at all useful

*Will you consider the Linked Web APIs dataset in near future?*

- Yes, I will consider the dataset already next week.
- Yes, I will consider the dataset already next month.
- Yes, I will consider the dataset in the following few months.
- No, I won’t.
A. Appendix

Page 4/6: Web API Provider?

Are you or your company (or project) providing an API?

- Yes (goto page 5)
- No (goto page 6)

Page 5/6: Web API Provider

How many APIs do you provide?

- 1 API
- 2 - 5 APIs
- 6 - 10 APIs
- more than 10 APIs

How do you measure the impact of your APIs?

- the number of clients using our API
- the number of API requests per day/week/month
- the amount of content processed per day/week/month
- we don’t measure it yet
- Other:

How do you compare the popularity of your API to the others APIs in the API ecosystem? How do you compare to your competitors?

- We manually collect information about APIs similar to ours.
- We analyze the consumption of ours and others APIs in an API and mashup repositories (e.g., ProgrammableWeb.com)
- We pay others for such research.
- We run surveys to collect this information.
- We don’t compare our APIs to others.
- Other:
How useful do you find the Linked Web APIs dataset from perspective of Web API provider? E.g. add descriptions of your APIs as part of the dataset and increase the visibility of your APIs, analyze the APIs of your competitors, mine the recent trends (protocols, formats, types of APIs), etc.

- Very useful
- Useful
- Somewhat useful
- Little useful
- Not at all useful

Will you consider the Linked Web APIs dataset in near future?

- Yes, I will consider the dataset already next week.
- Yes, I will consider the dataset already next month.
- Yes, I will consider the dataset in the following few months.
- No, I won’t.

Page 6/6: Web API Analysis

How important is for you to know the recent trends in the API ecosystem? E.g. what protocols and data formats prevail in the recent period, which industry sectors provided most APIs (tourism, commerce, enterprise, etc.).

- Very Important
- Important
- Somewhat important
- Little important
- Not important at all

Would you like to mine the dataset independently and build custom reports? E.g. what are the most popular protocols or data formats, what industry sectors provides most APIs, etc.

- Yes
- No
The Linked Web APIs dataset can be useful to...

- find and select relevant APIs
- increase the visibility of APIs into the API ecosystem
- track the popularity of the Web APIs
- compare our APIs to others
- evaluate the recent trends in the API ecosystem
- automated composition of Web APIs
- Other:
A.2 Annotation Guidelines: Entity Salience Dataset

In this section we present the annotation guidelines used for crowdsourcing the entity salience dataset (see Section 4.2.3). The crowdsourcing process was implemented and executed within the Figure Eight platform\(^2\) (previously known as CrowdFlower).

**Overview**

We kindly ask you to assist us in an experiment aimed at estimating the level of salience of entities in free texts.

**Instructions**

We provide

- A short text, and
- A highlighted entity in the text

Your task is to rate the level of salience of the highlighted entity for the article.

**How-To:**

- Step 1: Read the text.
- Step 2: Determine how salient is the highlighted entity for the text.

The entities can be classified as:

- Most Salient - Entities with the highest focus of attention in the article. The document is mostly about the these entities, or the entities play a prominent role in the content of the article.
- Less Salient - Entities with less focus of attention in the article. The entities play an important role in some parts of the content of the article.
- Not Salient - The article is really not about the entities.

Make sure before you start to understand the definition of a "SALIENT ENTITY" and how to rate the entity salience!

**What is a Salient Entity?**

Salient entities are entities which hold the focus of attention in the article. The article is usually about a small subset of salient entities.

\(^2\)https://www.figure-eight.com/
A. Appendix

Which entities are MOST SALIENT for an article?

The article’s content is mostly about these (most salient) entities. The most salient entities play a prominent role in the article. If you are asked to summarize the content of the article, you will definitely mention these entities.

Which entities are LESS SALIENT for an article?

The article is not really about these entities but they play an important role in some parts of the article’s content. The article has less focus of attention on these (less salient) entities. You can summarize the article also without mentioning these entities.

Which entities are NOT SALIENT for an article?

The article is not about these (not salient) entities. These entities appear in the article just to better describe the story presented in the article.

TIPS on how to identify candidates for salient entities in the text?

The answer of the following question can give you the initial set of possible salient entities for the article: About which entities is the article about?

Example #1

Consider the following text and the entity Avnet Inc. The task is to rate the salience of the entity Avnet Inc for the following article.

“Avnet Inc said it filed with the Securities and Exchange Commission a registration statement for a proposed public offering of 150 mln dlrs of convertible subordinated debentures due 2012. Avnet said it will use the net proceeds for general working capital purposes and the anticipated domestic and foreign expansion of its distribution, assembly and manufacturing businesses. The company said an investment banking group managed by Dillon Read and Co Inc will handle the offering.”

Solution: The article is primarily about the entity Avnet Inc (its a company) and its its strategic decision to submit for a public offering. Since the article is primarily about the entity Avnet Inc, we will mark the entity as "MOST SALIENT".

NOTE #1: If the task was to determine the salience of the same entity "Avnet" occurring in the second sentence, we will also mark this entity as "MOST SALIENT".

NOTE #2: If the task was to determine the salience of the entity “offering” (from the 1st sentence), we will most probably mark this entity as “MOST SALIENT”.

Be Careful Of: Entity salience is NOT SAME AS entity importance - Entity importance refers to importance outside of the scope of the document. For example, Michael Jackson
is a very important entity, but he might be not it the focus of attention in the article and therefore he will is not a salient entity for article.

**Summary**

Your task is to read the text and rate how salient is the highlighted entity for the article: "most salient" - the article is mostly about the entity, "less salient" - some parts of the article about the entity, or "not salient" - the article is not about the entity. Note, there are only few most and less salient entities in each article, while many not salient entities.
A.3 Linked Web APIs Example: The Twitter API

In the following listing we show an excerpt of the Linked Web APIs dataset for the Twitter API. Due to space constraints, we list only the most relevant information. For a complete resource representation, see http://linked-web-apis.fit.cvut.cz/resource/twitter_api.

```turtle
@prefix ls: <http://linked-web-apis.fit.cvut.cz/resource/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix prov: <http://www.w3.org/1999/02/22-prov#> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix lso: <http://linked-web-apis.fit.cvut.cz/ns/core#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .

ls:twitter_api a lso:WebAPI ;
  rdfs:label "Twitter API" ;
  prov:generatedAtTime "2006-12-08"^^xsd:date ;
  lso:assignedTag ls:microblogging_tag , ls:social_tag ;
  lso:assignedCategory ls:social_category ;
  lso:supportedDataFormat ls:atom_format , ls:rss_format , ls:xml_format , ls:json_format ;
  lso:supportedProtocol ls:rest_protocol ;
  dcterms:title "Twitter API" ;
  prov:wasAttributedTo ls:twitter.com_provider ;
  lso:rating "4.1" ;
  foaf:weblog "<http://blog.twitter.com/>" ;
  dcterms:description "The Twitter micro-blogging service includes two RESTful APIs. The Twitter REST API methods allow developers to access core Twitter data. This includes update timelines, status data, and user information. The Search API methods give developers methods to interact with Twitter Search and trends data. The API presently supports the following data formats: XML, JSON, and the RSS and Atom syndication formats, with some methods only accepting a subset of these formats." .
```

Listing A.1: Turtle representation of the Twitter API in the Linked Web APIs dataset.


12, 36, 68

12, 36, 58, 68

12, 36, 58, 68

12, 37, 63, 68

12, 36, 68

12, 36, 68

12, 36, 68

13, 42, 47, 73, 112
13, 42, 47, 73

13, 43, 73, 124, 125

13, 42, 43, 47, 73

13, 43, 47, 73

13, 44, 45, 109, 114, 120, 121

14, 49, 50, 51, 133, 146

14, 49, 50, 51, 133

14, 48, 50, 51


Bibliography

361219.361220
31

32

32, 33

33

33

41

41

41


Bibliography


83


89


89


89


97


103


103


103


