Approach to leverage Websites to Application Programming Interfaces through Semantics

Dissertation

by

Ioannis Stavrakantonakis

submitted to the Faculty of Mathematics, Computer Science and Physics of the University of Innsbruck

in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

supervisor:
Univ.-Prof. Dr. Dieter Fensel

Innsbruck, March 2017

Approach to leverage Websites to APIs through Semantics

by

Ioannis Stavrakantonakis

Submitted to the Faculty of Mathematics, Computer
Science and Physics of the University of Innsbruck,
in partial fulfillment of the requirements for the
degree of Doctor of Philosophy

Abstract

This manuscript describes a methodology designed and implemented to realise the recommendation of vocabularies based on the content of a given website. The goal of the proposed approach is to generate vocabularies by reusing existing schemas. The automatic recommendation helps to leverage websites to self-described web entities in the Web of Data; understandable by both humans and machines. In this direction, the implemented approach is wrapped within a broader methodology of turning a website in a machine understandable node by using technologies that have been developed in the scope of the Semantic Web vision. Transforming a website to a machine-understandable entity is the first step required by the websites side in order to narrow the gap with web agents and enable the structured content consumption without the need of implementing an Application Programming Interface (API) that would provide read-write functionality. The motivation of the thesis stems from the fact that the data provided via an API is already presented on the corresponding website in most of the cases.

Supervisor: Univ.-Prof. Dr. Dieter Fensel - University of Innsbruck
Co-supervisor: Ass.-Prof. Dr. Anna Fensel - University of Innsbruck
External supervisor: Univ.-Prof. Dr. Sören Auer - University of Bonn

To my parents Ioanna Kafkala and Nikolaos Stavrakantonakis, who taught me to work hard and with passion, and to my wife Giasemi and son Iasonas.

Acknowledgements

First and foremost, I'm indebted to my supervisor, Dieter Fensel, director of the Semantic Technology Institute of Innsbruck (STI Innsbruck), who has been supportive since I started working in the institute by exposing me to opportunities and challenges with significant potential in research, direct applicability on the Web and personal growth. Being one among the pioneers of the Semantic Web community, provided to me the privilege to discuss about how research evolved in the Semantic Web field and seek for feedback to my ideas from key members of it, either in conferences or various research events of the institute.

A special thanks to my co-supervisor Anna Fensel, who together with Dieter Fensel, guided me through all these years to shape my topic, reach my goal and contribute to the realisation of the vision of the machine-understandable Web. I wouldn't have pursued a PhD at the University of Innsbruck if she hadn't informed me about the position at the institute. Her feedback and input specially in the last period helped me to finalise the scope of the submitted work and overcome any issues.

Also, I would like to thank Sören Auer for his valuable feedback and for being my external supervisor.

Apart from the strong visionary side of Dieter Fensel and STI Innsbruck, another equally important aspect that made my experience in the institute unique, was the involvement in European projects of significant size with many stakeholders. I would like to thank Holger Lausen, the CTO of Seekda, for the fruitful collaboration and insightful discussions we had during my one month visit in the summer of 2013 at their premises in Vienna.

The presented PhD work has been partially supported by the Internet Foundation Austria in the scope of Netidee 2015 and in the form of a scholarship (Stipendium 1548) from November 2015 until the completion of the research and submission of the dissertation. Therefore, I'm really grateful to the Netidee and the scholarship program for the provided financial support in the second half of my PhD research endeavours.

Also, feeling grateful to the staff members of the STI Innsbruck for the great time we had as colleagues. I'm specially thankful to the other PhD students that we shared the same office and Ioan Toma for his guidance as the head of my research unit when I joined. It was a pleasure working together.

The two daily routines, working as a software engineer in a fast paced company like KAYAK and during the after work hours conducting the research for my PhD, were truly exhausting. It would not be possible to conclude this PhD if I didn't have the immense support and encouragement of my partner in life, Giasemi Morou, who has been to my side throughout this PhD since day one. Finally, the last mile towards the completion, wouldn't be the same without the smile of my little nine months old son, Iasonas.

The PhD was a life changing experience, pushed me to go beyond what I would believe it is possible to happen, and taught me to persist in pursuing of ideas that matter while keeping an eye to the cost of the invested time and energy.

Ioannis Stavrakantonakis
Berlin, March 12, 2017

Contents

List of Figures

List of Tables

List of Listings

List of Definitions

List of Algorithms

Chapter 1

Introduction

Towards a vocabulary terms discovery assistant

The Semantic Web technology stack has matured enough to consider the technology contribution to slide towards the "plateau of productivity" of the hype cycle for Web Computing. This term was coined by the information technology and research company Gartner to describe the period in the lifecycle of a technology when its adoption by the industry has started [51]. Main characteristics of the plateau include the need of research to bridge the gap between theory and application in practical and applicable ways; and the transformation of real world application needs into research directions. In this context, the research work behind this thesis started by exploring the Semantic Web horizon and continued any previous research directions in the related fields. The PhD journey brought contributions in a few different directions, though having a main target, i.e. the application of Semantic Web and facilitation of the uptake in areas that the results of the Semantic Web technologies can give better answers to Web Computing problems. Reading through the motivation in Section 1.1 it is prominent that the main reason that we moved to the Semantic Web, as presented at the beginning of this introduction, has not reached its full potential and transform the Web content into machine understandable content. Therefore, within this manuscript the main aim is to contribute to the bridging of the gap between the Semantic Web technology proposition and the application of it by facilitating to address the usability of ontologies (the term is used interchangably with the term *vocabularies* across the dissertation) for annotating Web content.

Thus, following the description of the thesis context, the rest of the chapter includes a presentation of the motivation behind the research work dimensions in Section 1.1; a brief introduction and declaration of the research thesis in Section 1.2; a summary of the contributions that took place during the research work in Section 1.3 and the structure of the disseration chapter by chapter in Section 1.4.

1.1 Motivation

The idea to build web agents to understand web content and interact with it in order to realise a given plan and achieve goals has been part of the Semantic Web vision [28] since the very first steps of the related working groups. As described by Hendler [39] with the example of intelligent travel agents, those systems should be *communicative, capable, autonomous* and *adaptive*. The various vocabularies and ontologies[1], that have been developed until today, contribute towards making web content machine-understandable through semantic annotations, which will enable web agents to behave as described before.

Exposing the content of a webpage in a structured way following a vocabulary it helps to easier communicate with partners as the vocabulary plays the role of a contract among the stakeholders. In this way, potential partners do not need to request a specific Application Programming Interface (API) from the webpage owner company in order to consumer their data, which leads to faster cooperations and cost reduction. In addition, search engine providers can make sense out of the published content by consuming the instances of the vocabulary classes that appear in the web page content as semantic annotations.

On the other hand, there is a significant difficulty to overcome related to the generation of semantic annotations. According to published surveys about the Web of Data and the semantic annotations coverage, it is unquestionable that there is a

[1] The terms ontology and vocabulary are interchangeably used across the manuscript.

long distance to cover in order to consider the Web sphere as a structured information datastore. According to the Common Web Crawler[2] only ca. 17% (2,722,425 out of 15,668,667) of domains were found with triples in Q4 2014. Additionally, the corpus analysis presented in [56] and [10], showcases that the growth rate of Semantic Web formats publised on the Web year over year could be considered significant, however, the question about the vocabulary discovery and application ease still remains unchanged. Focusing on specific domains within the scope of this thesis, we realise that local businesses in business areas, that heavily rely on the Web presence, are far behind the structured data paradigm and expectations. The ontology triples existence ratio is significantly low as presented in [85], as only 5% of the examined hotel websites include ontology triples. These reasons reside at the core of the thesis' motivation.

Moreover, investigating further, it is relevant for the motivation frame of the presented approach to analyse the level of misuse of vocabularies. Exposed structured data by using semantic annotations could be wrongly realised by using vocabularies in the wrong way or simply having wrong format. In both cases, the result would not be accessible by a search engine or other parsing process. Hogan et al. present in [42] some of the issues that they discovered while systematically examining existing Linked Data. They discovered many error types including syntax errors, some "ontology-hijacking" cases, which refers to misuse of the terms' semantics and ca. 15% of triples were using undeclared property URIs. Meusel et al. in [55] demonstrate their findings about common errors in deployed microdata that implement schema.org annotations and provide heuristics for fixing them at the consumer side. Proof of the complexity in creating annotations is the fact that according to their study over half of the examined sites (56.58%) confuse object properties (OP) with datatype properties (DP), at least once, by using OPs with a literal value instead of referring to an instance of a class.

Apart from the facts about wrongly applying the underlying technology and misusing the vocabularies, it is important to understand the benefits that semantic annotations bring to the Web. The combination of the misuse with the impact of annota-

[2] http://www.webdatacommons.org/structureddata/

tions will complete the view which lead to this thesis. Transforming the webpage to a machine-understandable resource, facilitates the understanding of the content by the search engines which enables the search engine's ranking algorithm to better serve the user with a consequence of better visibility for the website per se, as it will be ranked better in case the content is relevant to the user's search query. An example of improving the online visibility of local businesses by using semantic annotations is presented in [89], in which Toma et al. present the impact of weaving annotations to a touristic provider.

Taking the abovementioned facts in consideration, it is apparent that there is a strong need in assisting the data and content publishers with methodologies and tools not only to weave annotations and generate them in the correct way, but also to discover the appropriate vocabulary terms. Facilitating the various steps in the process of producing annotations, will drastically change the learning curve and the results of the process itself. In brief, the presented approach is positioned in the wider direction of facilitating the uptake of vocabularies and creation of structured data out of the already published content in the Web. Finally, strong aspect of the motivation to conduct reseach in this direction is the curated vocabulary repository called LOV [91], which reflects the status of the vocabulary space and the dynamics inside it.

1.2 Research thesis

Inspired by the related work in the field of Semantic Web and motivated by the problems that remain unsolved in realising the full potential of the technology stack developed through the years of the Web and the direction of Semantic Web, this section aims to outline the main research questions of the PhD. However, this section cannot comprehensively cover the contribution surface of the research conducted during the PhD. In this regard, Section 1.3 outlines the main contributions of the presented PhD work that support the title and main goal of it.

Several research questions have helped through the PhD journey to formulate the research thesis which is proved by the proposed approach. Thus, the main research axes of the thesis aim to answer the following questions:

How can a website be semi-automatically leveraged to a machine-understandable data interface?

The answer to this question is given in Chapter 5, which presents the framework workflow that starts with a given webpage as input and finishes with the result set of terms that can be used to annotate the webpage. In brief, the presented methodology combines the answers to the next two research questions in order to provide not a random set of terms that is relevant to the webpage based on text similarity, but the most optimal set that could be selected in an automatic fashion.

How can Linked Open Data support the choice of vocabulary terms to annotate a webpage?

Chapter 4 answers the question by demonstrating how to gather knowledge about the importance of the various vocabulary terms and the usage patterns of them. The existing entities in the Open Data sphere are used to extract the classes and properties of the vocabularies that are mostly used.

How can a vocabulary term be selected over one of the alternarives for a specific webpage content?

Answering this question required analysis of the dynamics as those are materialised in the graph of the Linked Open Vocabularies (LOV). LOV is considered the most comprehensive directory of vocabularies and various metrics are introduced in Chapter 4.3 to facilitate the ranking of the vocabularies and the terms. These ranking metrics are taken in consideration by the core algorithm to decide on the final vocabulary suggestion.

Wrapping up the above questions into one research thesis supported by the presented research endeavours, the thesis statement reads:

Vocabulary recommendations can be semi-automatically generated for a given webpage with a recall over 80% and it can outperform a manual selection of vocabulary terms.

Apart from the main contribution of the presented research work, we could realise the completed work as a set of contributions that support the vision of making the consumption of web content more accessible by Web agents, without the need of implementing a separate application programming interface (API) to expose data points, which are already published. In this direction, the next section puts the various contributions of the work on a “research map”, and also maps those contribution bits to the various following chapters and sections of this manuscript.

1.3 Contributions

The topic of the PhD thesis as it is reflected by the title of this manuscript, i.e. *“Approach to leverage Websites to Application Programming Interfaces through Semantics”*, contributes to the research field of Semantic Web and the development of a metadata layer on webpages. The main goal of the research is to provide methodologies that would allow websites to provide content that can be consumed by machines without the need to develop a separate Application Programming Interface (API) for that purpose. Therefore, in contrast to the development of an API the main idea involves the semantically explicit description of the represented website content. This aim is realised by introducing a semantic layer which allows to disambiguate the meaning of the presented data. Providing structured and semantically disambiguated data via the website, leverages it to a self-described API, which empowers any consumer of the website content, to extract information in a structured way equivalent to the output of separate API endpoints.

The previously presented research questions together with the thesis statement lead to the research in various directions that are combined together to address them and to evaluate the thesis statement. The following five contributions have been selected to represent the core research efforts in the scope of the PhD work.

- *Ranking of the vocabularies in the Semantic Web space.* The exponential grow of the vocabulary (ontology) space, as depicted by the Linked Open Vocabularies (LOV) curation directory, constitutes the need to assign a score to each entry to be able to sort them. Calculating a score for an individual within a set of many entities is a problem with many different solutions. The way the fomula for the scoring is designed should reflect the most important aspects that have been considered as the key factors in the context that the score will be used to compare the vocabularies. The proposed approach introduces a new dimension, i.e. the author list of the vocabulary.

- *Ranking of terms using Linked Open Data (LOD).* A vocabulary comprises a set of terms, with each one to have a different importance. Therefore, in addition to the ranking of vocabularies, the ranking of the terms is equally important in order to accomplish the selection of vocabulary terms given a set of keywords. In this scope, the presented approach combines the various sources of information about the usage of vocabulary terms in the LOD cloud and assigns a score to them in order to be later combined with the vocabulary scoring.

- *Recommendation of a set of vocabulary terms based on a keyword set.* The above two contributions are combined in order to semi-automatically provide recommendations for a webpage by proving the set of keywords that would better describe the page. In this way, the usage is considered semi-autonomous as the user of the approach needs to provide the most important keywords of the page and not only the webpage that contains them. However, this method allows a lower number of false positives in the result set of terms as it will be discussed later.

- *Design and development of an approach that combines Natural Language Processing (NLP) and other techniques with the results of the above mentioned directions.* The ultimate goal of the approach is to allow the users to discover vocabulary terms by giving only as input the target webpage. To achieve this aim, an NLP layer has been introduced to facilitate the extraction of keywords that later play the role of search tokens for vocabulary terms. In addition to the extracted keywords, the approach defines a set of rules that allow to extract other equally important parts of the webpage that need to be annotated and the NLP per se does not suffice.

- *Definition of a new vocabulary that facilitates the description of the query.* In the scope of representing the result vocabulary of the recommendation process, a technical vocabulary has been developed. The new vocabulary aims to facilitate the presentation of the keywords and the respective terms together with the final ranking score.The new vocabulary is named *vSearch* and described later in the manuscript.

The abovementioned contributions are presented throughout the sections as Section 1.4 describes. **The corresponding results have already been published in [89], [78], [79], [80] and [81].**

Another set of accomplishments, beyond the major contributions described in the previous paragraphs, include work under the umbrella of the Semantic Web. This work is related to data mediation ([31], [83], [84]), data retrieval ([85], [9]), data modelling ([77]), extraction of Social Web data ([82], [88]) and exploration of the multi-channel communication ([27]). Some of those contributions are discussed in Chapter 2, as part of the introduction to the Web of Data exploration. However, an extensive presentation of them is not part of the thesis scope as it focuses on the discovery of vocabulary terms for a given webpage rather than the rest of the research directions that were explored during the PhD time.

1.4 Thesis structure

The thesis contributions develop on an easy to follow presentation of the proposed approach starting from the status quo check, moving to the theoretical research and finishing through a showcase of applications of the methodology. In terms of structure, eight chapters follow the introduction, as summarised below:

- *Chapter 2 The Web of Data: Exploring the Semantic Web landscape* presents the various types of data that are available on the Web and how the Web of Data is formed based on hypermedia linking. Furthremore, it explains the representation of knowledge in the Web of Data by describing formats and vocabularies, discusses the exchange of structured data and the interoperability issues that can be addressed with semantic vocabularies.

- *Chapter 3 Related work and State of the art: Harnessing the power of the Web of Data* represents the conducted state-of-the-art survey in the field of semantic annotations, vocabularies discovery and vocabularies ranking.

- *Chapter 4 Approach outline: Towards the transformation of websites to APIs* examines the opportunities beyond the state-of-the-art and introduces the idea behind the proposed approach by discussing how far are the websites with the concept of machine understandable content and how could they play the role of an API for the already presented data of the website. dives into the dynamics of the existing vocabularies as those are curated by the Linked Open Vocabularies directory in order to establish methods to discover vocabulary terms applicable and suitable for the annotation of a given webpage.

- *Chapter 5 Approach and Methodology design: The LOVR framework* defines the architecture of the presented approach, presents the ranking algorithm that has been developed to facilitate the selection of the best terms from a list of result terms, as well as the algorithm to rank the existing vocabularies in the Linked Open Vocabularies graph. Additionally, the various vocabulary generation aspects are presented in conjunction with the vocabulary that was created to describe

- *Chapter 6 Approach implementation: Development reference and usage guide* is the most technical part of the thesis, as it includes the implementation details about the development of the proposed approach in the form of a Web Service.

- *Chapter 7 Use Cases* showcases the effectiveness of the approach by applying the methodology on three use cases from various sectors, i.e. a local business page, a recipe, an article, and a museum page.

- *Chapter 8 Evaluation* is responsible for providing the proof of concept for the proposed methodology by following two main evaluation scenarios. Initially, it presents a machine-based evaluation that basically evaluates the results of the LOVR framework on pages with annotations after removing them by comparing the generated annotations with the pre-existing ones.

- *Chapter 9 Conclusions and Future work* sums up in a few paragraphs the results of the conducted research and highlights the future directions.

The aim of this manuscript is to be understandable to a reasonable extent by people outside of the Semantic Web community as well. The developed prototype could be used in parallel in order to understand the outputs of the various modules as those are described through the chapters.

Chapter 2

The Web of Data

Exploring the Semantic Web landscape

Describing the Web from a data perspective can be approached in many different ways, while the most popular nowadays to be related to the depiction of the volume of the generated data and the velocity of content being published. Providing figures about the hours of videos produced hourly or number of pictures uploaded per minute is undoubtly astonishing. Historically, the Web started as a simple collection of HTML pages that included hyperlinks between them in order to refer to other sources of content. Later some meta tags, like the HTML *meta*, started to appear in order to facilitate the classification of webpages into category buckets and to empower the indexing in more efficient ways by the search engines.

Talking about data on the Web, it is important to clarify that it refers to anything published on the Web sphere, including text, multimedia, raw files, etc. Taking the previous note in consideration regarding the classification problem that the search engines were facing since their very first days, we easily realise the knowledge management issues that have arisen since the day that it was possible and easy for every user to publish data on the Web. Pure indexing for a keyword based search is hard to bring at the surface of the search results the most relevant ones as it is ignoring the context in which the content belongs to.

This chapter aims to describe the various dimensions of the Web of Data, the opportunities and the challenges in the exploitation of the data. It starts by presenting the presence of semantics on the Web in Section 2.1. The following section studies the information representation technology stack that is required to realise the semantic annotations concept in Section 2.2. In addition, Section 2.3 discusses the semantic annotation paradigm from the perspective of an API and how it can have a significant impact on the exchange of information in B2B relationships via the publicly shared Web content and documents.

2.1 Semantics on the Web

One of the simplest mechanisms, which is actually still being used, that facilitates the knowledge management of content, is the tagging system. Blogs (e.g. Wordpress), sharing platforms (e.g. del.icio.us, flickr) adopted the concept of tagging on the presented content in order to allow the users to explore the published data grouped in categories. Definition 1 gives a description about the core element of a tagging system, the *tag*.

Definition 1 (Tag) ***Tag*** *is a metadata keyword that helps to give information about the accompanied content by representing the category that it belongs to, or by highlighting the most important and representative term that appears in the content.*

Tags enable the transformation of local (contextual) category identifiers to URIs via the top level domain of the website that they appear in. In this way, they facilitate the access of the website content in a structural way without hierarchies. The tag concept is just one of the many paradigms that lie under the semantic annotations topic, which aims to give meaning to the Web of Data by explicitly annotating the published content. According to the above presentation of the tagging paradigm, it is apparent that both sides, the users and the online service providers, experiment with creative ways in order to manage through annotations the Web knowledge base that is constantly and exponentially expanding. The described need of managing the *Web of Data* leads to the principles of the structured data paradigm that empowers both users and applications to consume the published data with more effectiveness and ease.

Throughout this chapter there a lot of references to URLs and URIs. They are related to the URLs but also different enough and with special characteristics that allow them to differently serve the needs of the Semantic Web. Both types of identifiers have been fundamental building blocks of the Web infrastructure. In simple words, we could have them in mind as the medium to use unique identifiers for anything that has an online existence. Definitions 2 and 3 are derived from the famous Internet Standard RFC 3986 [8], which defines the details of the URI.

Definition 2 (URI) ***URI*** *stands for Uniform Resource Identifier and is a sequence of characters used to identify a resource. A resource could by anything and is not necessarily accessible via the Internet. Also, it is uniform by forcing a specific syntax, i.e. scheme ":" hier-part ["?" query] ["#" fragment], which provides a uniform semantic interpretation of different identifiers.*

Definition 3 (URL) ***URL*** *stands for Uniform Resource Locator and refers to a URI that in addition it provides means of locating the resource.*

Definition 4 (URN) ***URN*** *stands for Uniform Resource Name and refers to a URI that identifies a resource by name without providing location details or access methods.*

The various parts of the URI scheme are analysed under the RFC 3986 in [8], but for simplicity we could have in mind the simple example of an http scheme URI, like this: http://www.istavrak.com/data-driven-alerts#ifttt. In this example the scheme is “http”, the hier-part is “//www.istavrak.com/data-driven-alerts” and fragment is “ifttt”. The hier-part according to the standard consists of several parts; mainly the authority, which is “//www.istavrak.com” in our example and the path, which is the rest: “/data-driven-alerts”. However, it would be a mistake to associate the URI solely with the http scheme. Another example, from the daily things that web users interact with, is related to the URIs of a digital music service, Spotify. The

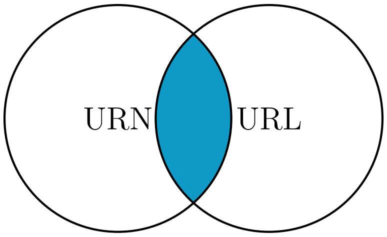


Figure 2-1: A URI could be a URL, a URN or both at the same time.

Spotify URI scheme is: $spotify :< artist|album|track >:< id >$ as described in the URI schemes directory in [46]. Thus, the URI of a track would look like this: *spotify:track:0gzqZ9d1jIKo9psEIthwXe* and it identifies the "U2 - Beautiful Day" track universally[1]. Thus, the Spotify example is a URN, while the website address example above is a URL. The website address explicitly specify that the access mechanism is the HTTP protocol with a specific network host/location, i.e. the top level domain name.

2.1.1 Website Data

In a broader scope all the Web data could be defined as website data. However, within this manuscript the term *website data* refers to the published content on websites of individuals, bloggers, local businesses, organisations, etc. excluding social networks and microblogging platforms (i.e. Twitter, etc.). The purpose of the websites in this category is mostly the presentation of information about the related entity (e.g. a hotel, the municipality of a region, a recipe, an article, etc.). Major asset in the content of a webpage is unquestionably the included text (more than 70% of the content for the majority of websites in terms of occupied space on the screen), but the most noticeable parts for the users are the visual elements like images and videos. Both categories of content are meaningless for the search engines that are indexing the published data, unless a smarter interpretation is in place. Apparently, the search engines have developed smart algorithms that try to implicitly acquire insights about the presented content, both text and visual based. For instance, a webpage that

[1] One can access the related album and artist URIs by using the corresponding API call: https://api.spotify.com/v1/tracks/0gzqZ9d1jIKo9psEIthwXe

includes text about a specific car model would implicitly mean that any included images within the body of the content (excluding any advertisement or other images that appear across all the pages of the website) have a great probability of depicting the car model that the text is describing.

All the above described example of content interpretation by a search engine indexing algorithm has a main drawback, i.e. the probability dimension regarding the **implicit extracted insights through text based reasoning**. Following up the car model example, we could imagine that the webpage text mentions some five digits numbers with a currency sign attached at the beginning or at the end of the digits (depending on the locale the currency is placed differently at a price presentation). Thus, a simple reasoning would be that the number refers to the price of the car at the retail market. Although it seems to be reasonable and with a great probability of correctness, there is also a significant probability that the price is irrelevant completely to the assumption that we already made. Could not be the case that the price refers to another car that is being compared with the car of the article or that it refers to the price of the same car but last year's model? Actually, only our imagination can narrow the number of alternatives that a price could refer to.

Therefore, a paradigm that lowers the borders of content interpretation by enabling an **explicit specification of the mentioned entities** would enable the better indexing of the data that would lead to better search queries answering at the search engines that we use. This paradigm, namely the semantic annotations paradigm, is analysed later in Section 2.2 from the prism of information representation. Thus, the annotated content can eliminate the obstacles of leveraging the content to actionable resource by machine based agents. The concept of annotating specific parts of the presented information with meta-information is called "Semantic Annotations" as defined in Definition 5.

Definition 5 (Semantic Annotation) ***Semantic Annotation*** *is a piece of metadata for an informational element that appears in a specific document and it is machine interpretable.*

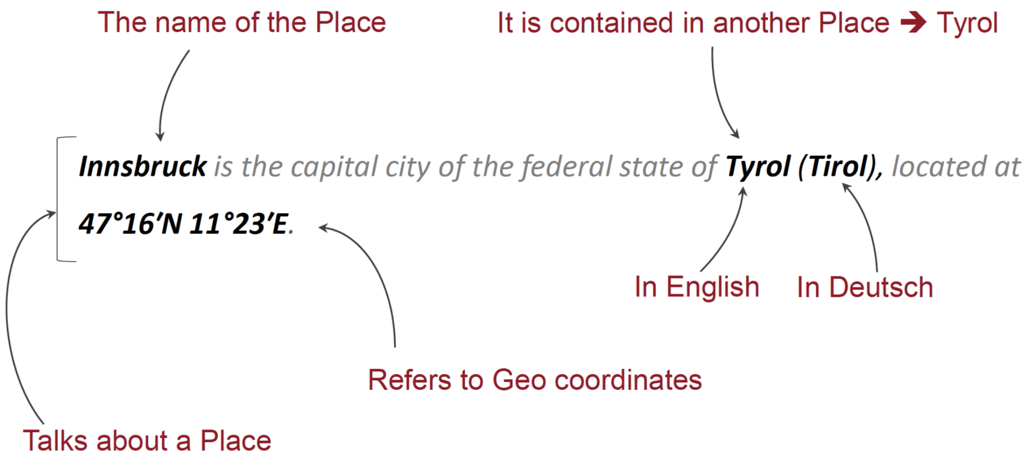

The name of the Place
It is contained in another Place ➔ Tyrol
Innsbruck is the capital city of the federal state of Tyrol (Tirol), located at
47°16′N 11°23′E.
In English
In Deutsch
Refers to Geo coordinates
Talks about a Place

Figure 2-2: Conceptual example of Semantic Annotations on a textual presentation type of information. The depicted text could be part of a webpage. The parts marked with bold are the referred text of the annotations, the red text is the meta-information of the annotation, and the arrows specify the referred text for each of the annotations. Furthermore, the words "Place", "Geo coordinates" refer to specific vocabulary terms (or ontology classes).

A conceptual example of what parts of text could be annotated is given in Figure 2-2, while listings 2.1, 2.2, 2.3 and 2.4 are examples of semantic annotations in various formats and using various vocabularies. In the example, we can see how it is possible to define what the depicted numbers refer to, what the whole document refers to and what the name of the place described in the document is. Furthermore, it is possible to specify a different name for each language and also we could link the document to another Web entity (via a URI) that defines the described place. In this particular case we could refer to the corresponding DBpedia URL[2] for Innsbruck. The added value of having the semantic annotations together with the content, allows the explicit specification of the content meaning and does not limit the metadata to an abstract statement about the main topics of the page.

On this axis a study was conducted in the PhD thesis scope to investigate the usage of annotated content and structured data of hotel websites in a specific region. The study is published in [85]. The seeds base was built by collecting data from various sources (Google Places, TripAdvisor) about the hotels that exist in Austria.

[2] DBpedia article for Innsbruck: http://dbpedia.org/page/Innsbruck

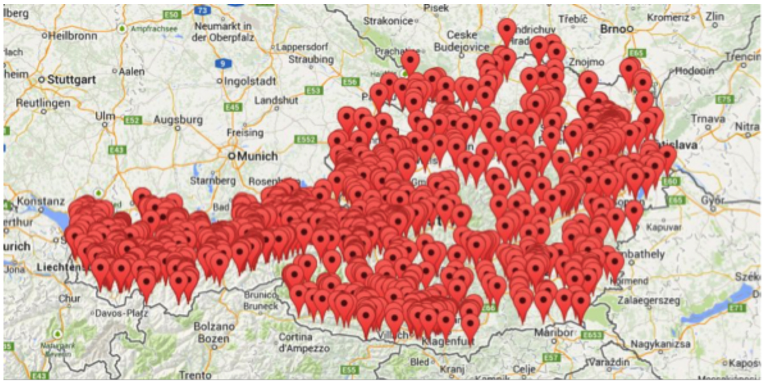


Figure 2-3: Geographic distribution of hotels in the examined dataset (published in [85]).

The seet dataset included hotel URLs, star ratings and geo-coordinates. Afterwards, a studying sample was selected randomly. Over 2000 hotels were selected, and 75% of the hotels selected had 3 to 5 stars and inside the borders of Austria as shown in Figure 2-3. The research workflow continues with the specification of the criteria with which the dataset was evaluated. The major subset of the criteria refer to the Semantic Web technologies that a hotel could apply on its website in order to gain more visibility on the web. Particularly, it focus on the existence of semantic annotations that could bring a hotel's website into a better position in search results of the major search engines (e.g. Google, Bing, Yahoo!, etc.) and with a richer presentation among the results by exploiting the opportunities that exist in the user interfaces of the search engines, such as in the case of Google with the Google Rich Snippets [86].

The next step towards gathering the hotel websites' data was the implementation of a Web crawler that would access the hotels' websites and extract information about the criteria that had been specified. The Web crawler is built in Python based on a popular open source high-level screen scraping and web crawling framework, i.e. Scrapy[3]. It is worth mentioning that a limitation at the depth for the crawler was set to three levels from the starting page in order to achieve better performance. However, this limitation cannot affect the accuracy of the results as the criteria are met at the very first pages of the websites in case they are met at all. The criteria were

[3] Crawling framework Scrapy: www.scrapy.org

divided between formats and vocabularies. Formats, described later in Section 2.2.1 refer to the technologies with which a web developer could add semantic annotations to a website (e.g. microformats, microdata, RDF) and vocabularies, described later in Section 2.2.2 are the sets of terms that can be used to annotate online content. The results of the analysis prove that most of the hotels completely ignore the existence of technologies that could enrich the website content with high level metadata and give machine readable meaning to the presented information. Only 5% of websites employ some Semantic Web technologies, while the rest seem to ignore the potential of adding semantics to their websites. The analysis concludes that the slow semantic technology uptake is hindered by both technical (e.g. difficult integration due to the usage of heterogeneous CMSs) and educational factors (e.g. knowledge about the new technologies and understanding of their advantages).

This outlook motivated the presented PhD work towards the direction of supporting the semantic annotation process, which leads to the transformation of content rich websites into machine understandable resources or in other words APIs.

2.1.2 Linked Open Data

Similar to the principles of the Open Source software licences, Open Data is meant to be freely shared by the owner with the users of the Web for access and use for any possible reason, i.e. commercial, research or personal. The principles of the initiative behind Open Data is briefly summarised by the Open Knowledge Foundation Network (OKFN) as: "*Knowledge is open if anyone is free to access, use, modify, and share it — subject, at most, to measures that preserve provenance and openness.*"[4]

According to the Open Data Handbook, Open Data is defined as: "*Open data is data that can be freely used, re-used and redistributed by anyone - subject only, at most, to the requirement to attribute and share alike.*"[5]

In principle, when talking about Open Data, we tend to refer mostly to government data as governments, including both ministries and municipalities, are those that are

[4] Open Definition: http://opendefinition.org/od/index.html

[5] Open Definition: http://opendatahandbook.org/guide/en/what-is-open-data/

in the possession of data resources that could be useful to leverage and consume within applications or various analyses. As it is already well described by OKFN in the respective online reference[6], Open Government Data empowers the society with transparency, added value in both social and commercial directions, and participatory governance; with the latter to mainly stand for the idea to encourage the citizens into real engagement with the governance layer by both consuming data in a waterfall manner but also contributing to it, or as it is literally stated *"making a full read/write society"*.

In this context, the Accessible Vienna Web application[7], which was implemented as part of the research endeavours of the presented PhD work, aims to bridge the gap between the Open Data[8] of the Vienna municipality[9] and the Google Maps Places data[10]. The main objective of the Accessible Vienna application is to support citizens or travellers with special needs in the city of Vienna[11]. It is not only the infrastructure of the city that is important for their daily activities but also the information about the various places (e.g. restaurants, cafés, theatres) and public facilities (e.g. parking places, subway stations). Accessible Vienna aggregates this information with details from Google Places on a map in order to enable users to find easily an accessible place that fulfils their needs. At the implementation level, we can see that the application links the Open Data with data retrieved from the Google Places API regarding details about the accessible places (i.e. photos, ratings, website, Google+ and opening hours). Thus, the user is enabled to both choose an accessible place like any citizen by checking photos, ratings and other venue related information, and find information about the availability of the required public services (i.e. parking spaces, accessible subway stations)."

Various challenges emerged through the implementation phase, including the mapping of the various entities described in the Open Data datasets with the Google

[6] Open Government Data: http://opengovernmentdata.org/

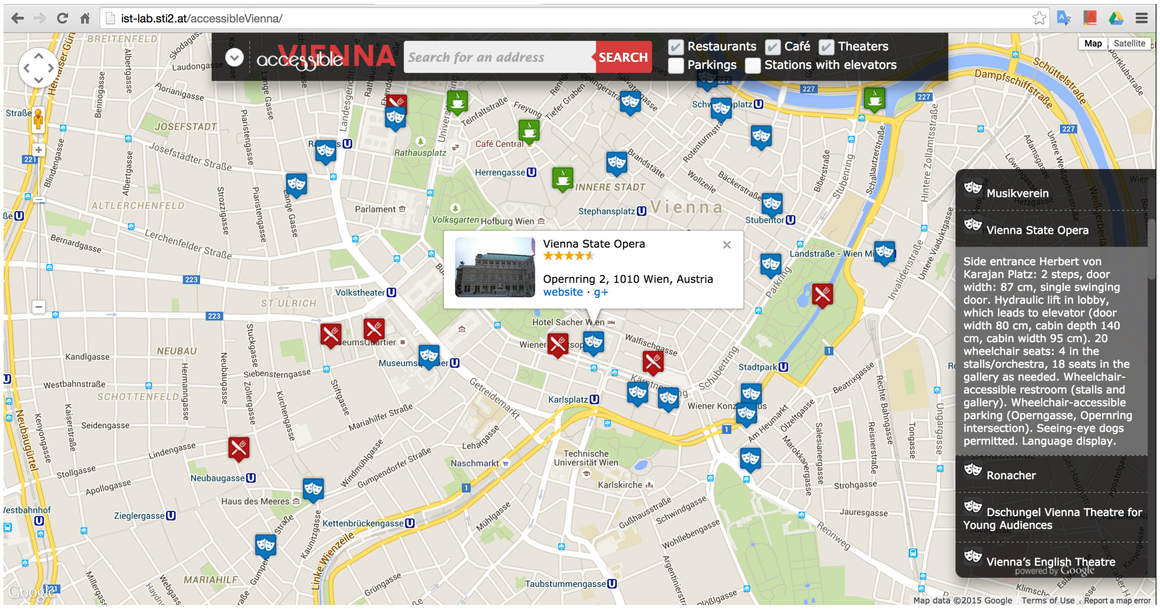
[7] Available at http://ist-lab.sti2.at/accessibleVienna

[8] Open Data for Austria: https://www.data.gv.at/

[9] City info: https://www.wien.info/en/travel-info/accessible-vienna

[10] Google Places API: https://developers.google.com/places/

[11] Teaser video: https://www.youtube.com/watch?v=AWblyobrJDk



Vienna State Opera
★★★★★
Opernring 2, 1010 Wien, Austria

Side entrance Herbert von Karajan Platz: 2 steps, door width: 87 cm, single swinging door. Hydraulic lift in lobby, which leads to elevator (door width 80 cm, cabin depth 140 cm, cabin width 95 cm). 20 wheelchair seats: 4 in the stalls/orchestra, 18 seats in the gallery as needed. Wheelchair-accessible restroom (stalls and gallery). Wheelchair-accessible parking (Operngasse, Opernring intersection). Seeing-eye dogs permitted. Language display.

Figure 2-4: Screenshot from the Accessible Vienna Web application.

Maps Places, e.g. theatres like the Vienna State Opera. One possible solution to the above mentioned problem is the annotation of the Open Data entities with their widely accepted unique reference identifiers (or as they are officially called Uniform Resource Identifier), URIs. Those URIs are in most of the cases taken from free and open directories that are dedicated to provide knowledge management solutions, like DBpedia or Wikidata. From a technical perspective, this enhancement would simply require the addition of an instance of the OWL property *owl:sameAs*, which would point to the respective URI. In our example, it would be the concept URI from Wikidata: https://www.wikidata.org/wiki/Q209937 or the URI from freebase: https://www.freebase.com/m/021c8v. Wikidata aims to provide a free knowledge base that both users and bots contribute to it as briefly explained by the Wikidata creators[12], while freebase is a structured representation of Wikipedia curated by Google and mainly used in the Knowledge Graph [11]. In the same fashion, the Google Places dataset would need to have references to URIs that could be used to automatically map the data entries with any other data entity on the Web. To support the merging of information bits from the two data sources, a new point of

[12] Wikidata introduction: https://www.wikidata.org/wiki/Wikidata:Introduction

Attribute	**Data Range**	**Data Source**
name	string	*Open Data*
address	string	
accessibility info	string	
coordinates	geo-coordinates	*Google Places API*
review score	double	
Google+ profile page	URL	
website	URL	
picture	URL	

Table 2.1: The model for the place entities in the Accessible Vienna Web application.

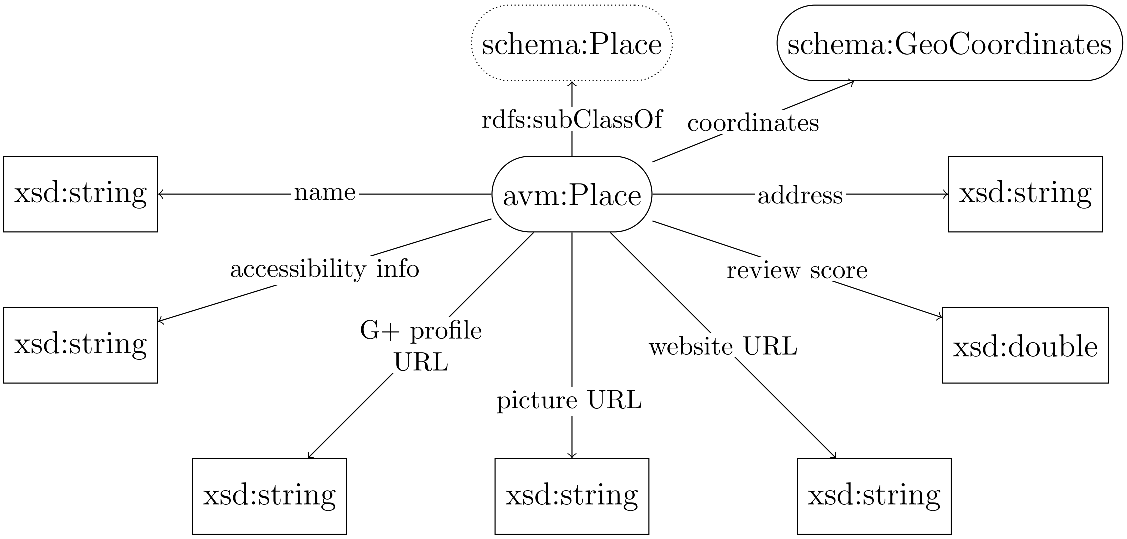


Figure 2-5: The model (avm) for the place entities in the Accessible Vienna Web application.

interest model was required that would combine the attributes of the two separate models that are followed by the Open Data source and the Google Places API. This is a simple model, subset of the schema:Place (https://schema.org/Place), but still with some additional attributes to accommodate specific data elements that are coming from the two data sources, like the Google+ profile URL. The model attributes are described in Table 2.1 and in Figure 2-5. Another approach could have in the model the Google+ profile page URL as a “sameAs” property of the “schema:Thing” class.

The presented application qualified as one of the fifty eligible applications of the Google Places Challenge in 2012[13] and is also listed as one of the many available appli-

[13] Google Places API Challenge candidates: http://googleplacesapichallenge.blogspot.de/2013/01/5-more-days-to-vote-for-peoples-choice.html

cations that are leveraging Open Data of Austria and Vienna, respectively mentioned at the Open Data portal of Austria[14] and the Open Government Wien portal[15]. A fast overview of the idea in a 60 seconds animation is available online[16] to watch and get aspired for more and even better use cases of Open Data.

Back at the Accessible Vienna application presentation, we discussed about the different datasets mapping issue, which apparently is one of the major cases that ontologies facilitate to solve by providing a data mediation layer. In the previous example, the middleware layer would be the data mappings to broadly accepted entity URIs. A detailed presentation of the ontologies is presented later in Section 2.2.2. Exploring the dimension of mapping the same entity identifiers through a semantic layer, one will wonder if we could refer to entities in a broader way within existing datasets. In simple words that would mean the inclusion of links to data objects within a dataset. These thoughts brought the Semantic Web research community to the definition [7] of the Linked Data as those datasets that conform to the following four rules:

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

On top of that, if the dataset is released under an open licence then it is recognised as Linked Open Data and is part of the Linked Open Data cloud (see Figure 2-6 and Figure 2-7). The importance of publishing data as Linked Open Data is the added value that the interconnection between entities bring to the Web surface. Consuming an entity described within the LOD ecosystem opens up the possibility to make use of the linked entities as well. The two figures of the LOD cloud, which were mentioned

[14] Open Data Austria: https://www.data.gv.at/anwendungen/accessible-vienna/

[15] Open Data Vienna: https://open.wien.gv.at/site/accessible-vienna/

[16] Overview video of Accessible Vienna app: https://www.youtube.com/watch?v=AWblyobrJDk

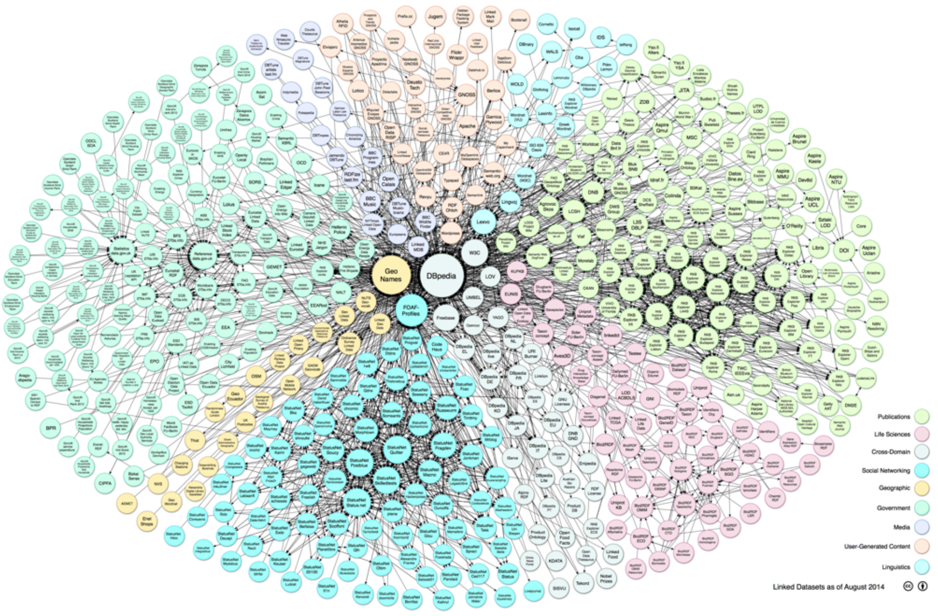


Figure 2-6: Linked Open Data cloud snapshot of April 2014 as shown on http://lod-cloud.net and captures the crawled dataset of the Linked Data Web [73]. The biggest node in the middle is the DBpedia node.

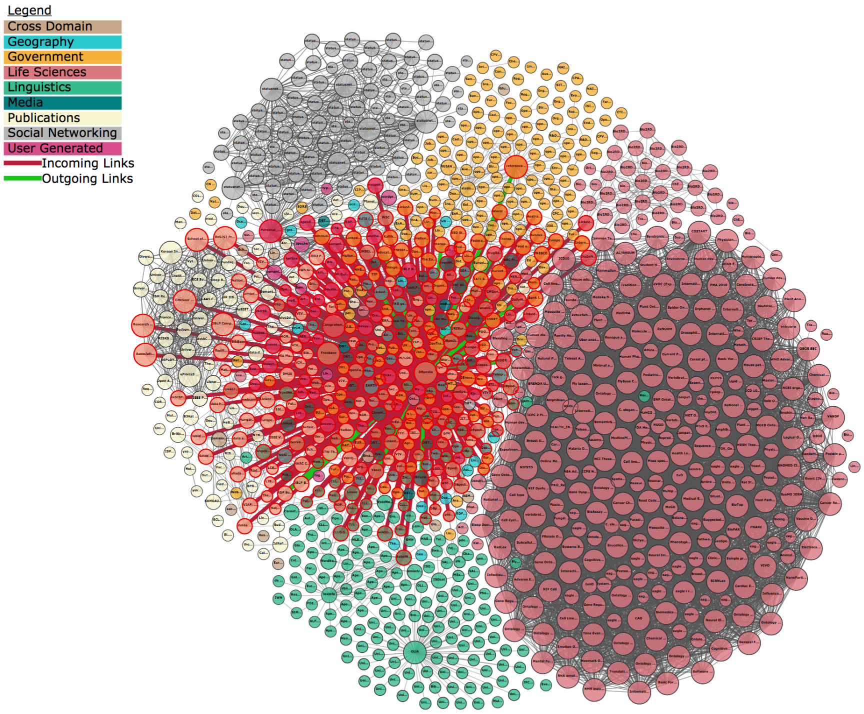


Figure 2-7: Linked Open Data cloud snapshot of February 2017 as shown on http://lod-cloud.net [1]. Based on metadata collected and curated by contributors to the Data Hub (https://datahub.io/). The DBpedia node has been highlighted in the middle together with the incoming links to it.

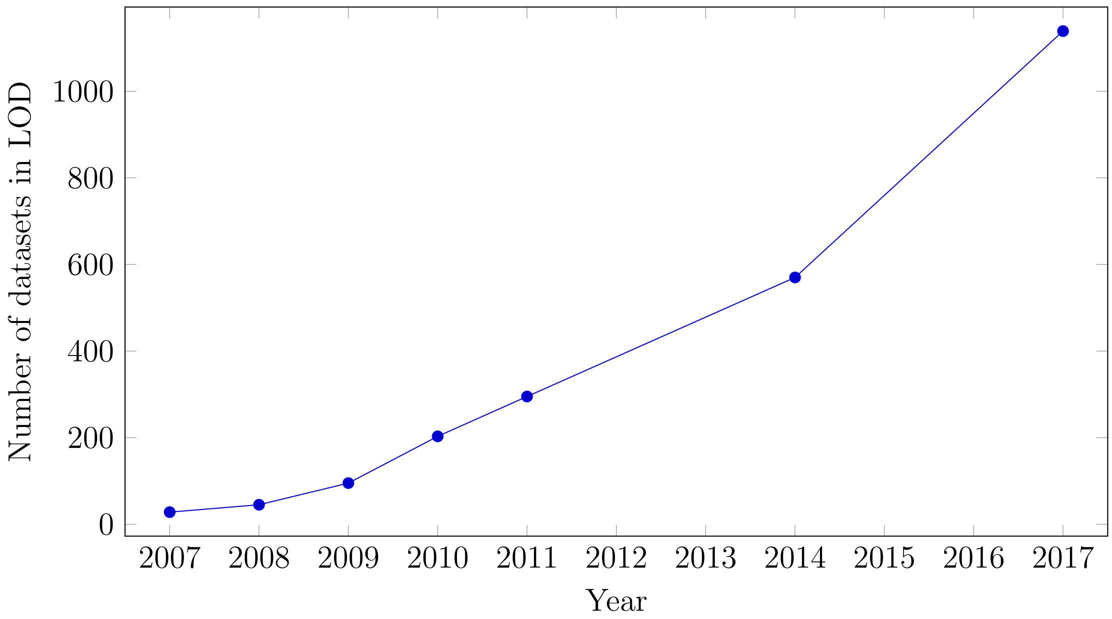


Figure 2-8: Growth of the Linked Open Data cloud in number of datasets involved. The number of datasets has been obtained from http://lod-cloud.net/#history, and for each only the latest has been taken in consideration.

before, include only datasets with more than 1000 triples and with connections to other datasets with at least 50 edges. The reader can realise the growth that has occured through the last few years in the direction of Linked Open Data generation by studying the trend line of Figure 2-8. In addition, visually comparing the snapshot of 2014 with the shapshot of 2017, we realise that DBpedia is not anymore the biggest node by being surrounded with many more and especially from the life sciences domain. A significant part of the diagram is also occupied by the government data, which is a great online database for applications that support the citizens' daily life, like the Accessible Vienna that was presented earlier (Figure 2-4).

According to the 3rd rule described above, the URI entity will give access to structured data formulated using the various standards. In this respect, the data producer needs to decide on a vocabulary that will be used to describe the published information in a structured way. Services that assist the exploitation of LOD by extracting various insights are presented later in Chapter 3. In the same direction, the proposed approach by this PhD thesis benefits by the existence of the LOD datasets to produce recommendations.

2.2 Semantic annotations' building blocks

In the previous section, the aim was to outline the data map of the Web in nowadays. One of the important assets of the Web is the aforementioned Linked Open Data paradigm. There are some fundamental concepts that enable this paradigm to be realised at the Web scale, besides the Web basic structures (e.g. URIs). Specifically, the Web of Data has been designed to grow on top of a stack of technological enablers that facilitate the ultimate goal of enriching the content with machine interpretable meaning.

The world of Web developers is familiar with the "metadata" concept in the HTML pages. The "meta" HTML tags were massively used since the early days in order to give some extra information to the search engines about the information that the user readable version included. Though, for many years the most popular "meta" tag used to be the "keywords" one, which was meant to be a list of words that reflect the content of the presented content to the user. The search engines would store those keywords for each crawled website and at a user search level they would be leveraged to a ranking approach in order to provide a sorted list of URLs that would better match the user's query. Not surprising, it was meant to be the most important of the "meta" elements and a whole web development industry dimension started to flourish by providing ranking optimisation services for search engines, the popular Search Engine Optimisation (SEO) process.

This first notion of explicitly defining the meaning of a webpage could be considered as the first attempt to realise the Semantic Web vision, however, it also differs in basic details, like the fact that the "keywords" or the other meta elements provide information about the webpage as a whole document rather than explicitly annotating parts of the document.

Assuming that the text of Figure 2-2 (*"Innsbruck is the capital city of the federal state of Tyrol (Tirol), located at 47.263479, 11.346044."*) appears on a webpage, the question then arises, what is it needed, from a technical perspective, to define the annotations on the webpage that explain the content? To answer this question two aspects need to be addressed: a) the format in which the annotations will be built and added to the webpage; and b) the language that will be used for the annotations. The various formats that can be used are presented in Section 2.2.1, and the language refers to the vocabularies, which are discussed in Section 2.2.2, respectively.

2.2.1 Formats

The purpose of this short sub-section is to briefly introduce or remind the various formats that have been developed in the Semantic Web ecosystem and play a crucial role in the implementation of semantic annotations within the content publishing on the Web. The formats described below provide a specific set of attributes that could be used together with domain specific vocabularies (explained in the second half of this section, Section 2.2.2) in order to annotate Web content. As the presentation of the formats evolve in the section, the reader is invited to realise the pros and cons for each one through the corresponding examples and explanations.

Microformats

Microformats[17] (abbreviated as μF) are conventions used to describe a specific type of information on a web page (e.g., people, organisations, locations, etc.). In general, microformats overload the class attribute in the HTML tags to assign descriptive names to entities. They can be realised as format and vocabulary combined. The second version of microformats, microformats2[18], adds prefixes to the terms in order to understand which class names are used by microformats, e.g. the root classes start with “h-” (h-review), simple text properties with “p-” (p-name), u-photo is used to annotate the URL of a photo, etc. Still the big disadvantage of microformats is the fact that the vocabulary is embedded together with the format. The class attributes are used by the Cascade Style Sheet (CSS) to give formatting to the layout of the page, which makes the microformats to look not ideal for weaving semantics to the content of a page. An example of a metadata block using microformats is shown in

[17] http://microformats.org/

[18] http://microformats.org/wiki/h-review

Listing 2.1.

```
<div class="hreview" id="hreview-Great-place!">
  <h2 class="summary">Great place!</h2>
  <abbr class="dtreviewed" title="2008-01-23T17:47">
         Jan 23, 2008</abbr>  by
  <span class="reviewer vcard">
        <span class="fn">anonymous</span>
  </span>
  <span class="type" style="display:none;">place</span>
  <img alt="..." src="http://..." class="photo">
  <div class="item">
    <a class="fn url" href="http://...">Cafe VI</a>
  </div>
  <blockquote class="description">
      <abbr title="5" class="rating">*****</abbr>
      A great choice to have lunch.
  </blockquote>
</div>
```

Listing 2.1: Example of semantic annotations in microformats - a place review with microformats hreview attributes.

As it was said before, microformats are not separating the concerns and in addition introduce a degree of complexity. For example at line 8 of the Listing 2.1, the *span* element is created in order to define the type of the annotated entity, however, as we can see at the same line, there is an inline *style* element, which forces the element not to be displayed to the final presentation when the website will render on the user's browser. The semantic annotations layer should not intervene between the underlying data and the final page presentation, but it should be part of the data layer or a separate dimension that will not affect the behaviour of the HTML markup elements.

RDFa

The Resource Description Framework in Attributes (RDFa) format provides a set of markup attributes to augment Web page content with semantic annotations. RDFa are based on attributes by re-using HTML tags and defining namespaces in the XHTML to assign types and names to entities and properties. None of the attributes introduced or used by RDFa have any effect on the rendering of the web page.

```
<div property="review" typeof="Review">
  <span property="reviewRating">
        5</span> stars -
  <span property="name}">Great place!</span>
  by <span property="author">Anonymous</span>,
  Written on
  <meta property="datePublished" content="2015-03-12">
        May 4, 2006
  <span property="reviewBody">
    A great choice to have lunch.</span>
</div>
```

Listing 2.2: Example of semantic annotations in RDFa - a place review with the RDFa property, typeof, content attributes.

Microdata

Microdata specification is similar to microformats, but introduces new HTML tag attributes (i.e. itemscope, itemprop, itemtype, etc.) that can host terms from any vocabulary. It is supported by schema.org and is part of the HTML 5 specification. In comparison to the aforementioned formats, we could say it combines ease of use, effectiveness and flexibility, all of which make it a great option for semantic annotations.

```
<div itemprop="review" itemscope
        itemtype="http://schema.org/Review">
  <span itemprop="reviewRating">5</span> stars -
  <span itemprop="name">Great place!</span>
  by <span itemprop="author">Anonymous</span>,
  Written on
   <meta itemprop="datePublished" content="2015-03-12">
   May 4, 2006
  <span itemprop="reviewBody">
   A great choice to have lunch.</span>
</div>
```

Listing 2.3: Example of semantic annotations in microdata and HTML5 - a place review with the microdata itemprop, itemscope, itemtype and content attributes.

JSON-LD

The JSON-LD[19] format could be interpreted as a Linked Data layer over the popular JSON (JavaScript Object Notation) data-interchange format. In a simplified way, we could describe the LD (Linked Data) as the layer that provides the needed information about the vocabularies and the namespaces of the JSON name/value pairs that appear in the body of a JSON object.

```
{
  "@context": "http://schema.org",
  "@type": "Review",
  "author": "Anonymous",
  "datePublished": "2015-03-12",
  "name": "Great place!",
  "reviewBody": "A great choice to have lunch.",
  "reviewRating": {
```

[19] https://www.w3.org/TR/json-ld/

```
        "@type": "Rating",
        "ratingValue": "5"
}
```

Listing 2.4: Example of semantic annotations in JSON-LD - a place review with the JSON-LD context and type attributes.

Examining Listing 2.4, we realise the straightforward way that the data is mapped to the annotation terms. The *@type* property is used to define the type of the object that is described from the rest of the properties at the same level, e.g. the Rating in our example. Furthermore, the *@context* helps to specify the namespace under which those types can be found. For example, the *ratingValue* is part of the vocabulary type *schema:Rating*, where *schema* is the namespace with URI http://schema.org.

The two mostly used formats, RDFa and microdata extend the HTML markup syntax in order to provide the needed prerequisites for adding metadata in a webpage. Microformats is a special case as it reuses the HTML class attributes and introduces specific class names in order to give meaning to the corresponding HTML tags, which makes it look like mixing the vocabulary semantics with the format itself. Last but not least, the JSON-LD format inherits the readability and simplicity from JSON and adds a few object members in order to define the type that describes the properties used at the left part of the surrounding pairs.

According to Bizer et al. [10], less than 7% of websites across the 40.6 million websites of the Common Crawl[20] in the index of 2012 include some data of the three main formations, i.e. RDFa, Microdata and Microformats. The distribution looks like this: RDFa 1.28%, Microdata 0.35%, Microformats 4.45%. Comparing the percentages to the figures for 2015 provided by the Web Data Commons[21], we realised that the distribution has changed to: RDFa 3.6%, Microdata 7.6% and Microformads 8.2%. Therefore, there is a significant increase in the usage of Microdata mainly because of

[20] http://commoncrawl.org/the-data/get-started/

[21] http://webdatacommons.org/structureddata/2015-11/stats/stats.html#results-2015-1

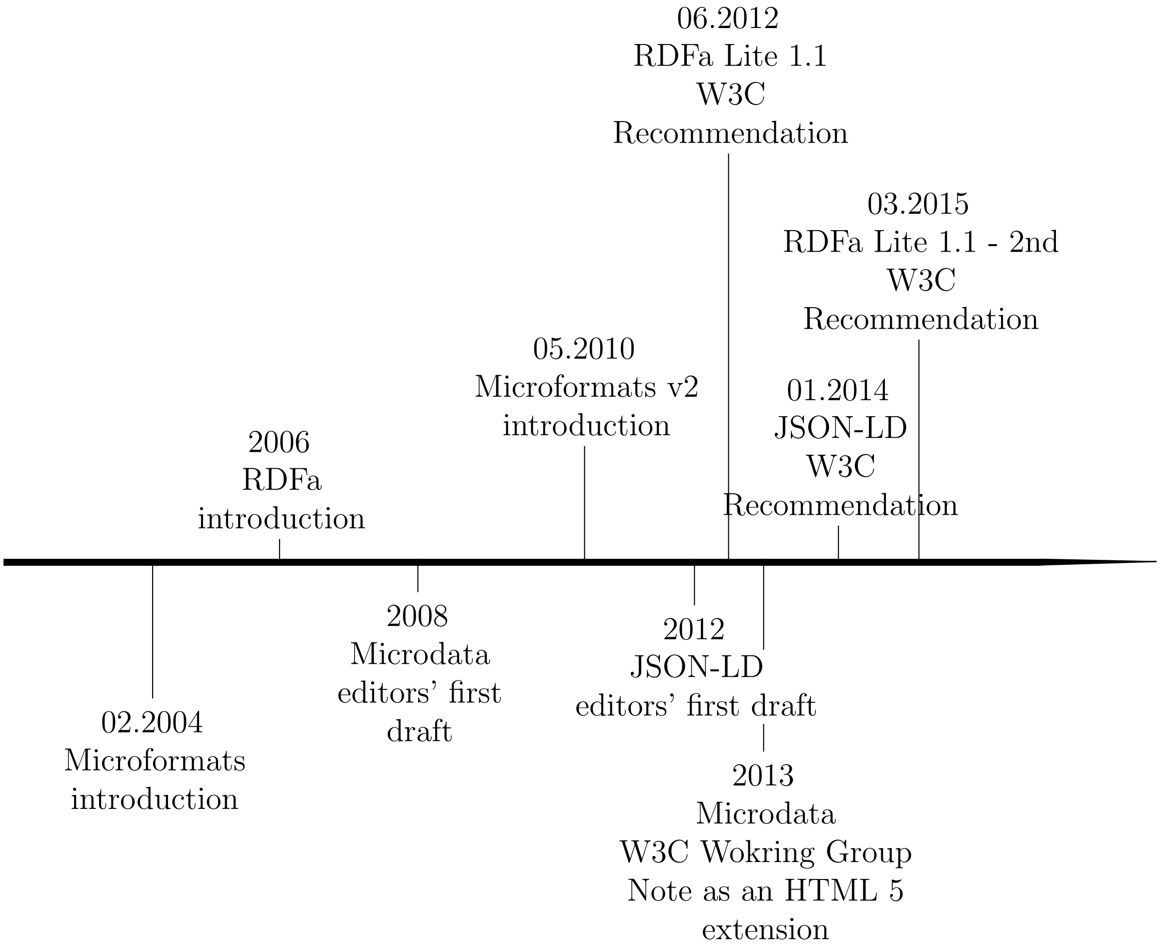


Figure 2-9: The formats that facilitate the inclusion of Semantic Annotations in HTML as emerged and developed through the years. The only approaches that have been leveraged to a W3C Recommendation are JSON-LD and RDFa Lite. However, all of the formats listed on the timeline are valid ways of weaving annotations on Web content and search engines are aware of all of them at the crawling level.

the popularity of the schema.org initiative the last few years. This is proved by the list of the top classes across the gathered data. Across the 20 classes, the 19 classes are from the schema.org collection of vocabularies. On the other hand, RDFa seems to be a complex solution and that is reflected to the slow adoption of the technology according to the usage figures.

2.2.2 Ontologies - Vocabularies

Ontology could be considered as the cornerstone of the Semantic Web due to its impact in modelling the world. Ontology is a term that was coined in [33] by T. Gruber and describes it as an explicit specification of a conceptualisation; leveraging the philosophical term to the computer science field. Therefore, the ontology is considered to be a specification of the language about a concept either in real or virtual life. Throughout this manuscript the words ontology and vocabulary are used interchangeably and considered as the same, which is also the direction of the W3C consortium[22].

Taking that starting point, mostly for clarity, Definition 6 defines the main characteristics of a Vocabulary while Definition 7 specifies the Vocabulary Term, which is the focal point in most of the sections of the thesis.

Definition 6 (Vocabulary) ***Vocabulary*** *is a set of labeled nodes that form a graph. The edges define the relationships among these nodes but also with nodes from external vocabularies.*

Definition 7 (Vocabulary Term) ***Vocabulary Term*** *is every node or edge of a Vocabulary, as that was defined in Definition 6. It consists of a label and iff it is an edge then it has also specified domain and range.*

The vocabulary term is synonym to both the property and the class as those are defined in the Web Ontology Language (OWL) recommendation[23]. Therefore, a term could be considered either an Object Property or a Datatype Property depending on

[22] https://www.w3.org/standards/semanticweb/ontology

[23] https://www.w3.org/TR/owl2-new-features/

the range of it. When the term's range is a node then it is equivalent to an Object Property, while in the case of literals it is equivalent to the Datatype Property. In sake of simplicity all the formulas and descriptions throughout the thesis refer to both property types and to the classes as vocabulary terms. The vsearch vocabulary depicted in Figure 5-5 in a later section is a good example of the semantics of the term that was previously specified. For example, the class vsearch:Query and the properties vsearh:keyword and vsearch:hasResultTerm are all considered vocabulary terms of the vsearch vocabulary. Specifically, vsearch:keyword is a Datatype property as it has range the XML Schema String type. However, the vsearch:hasResultTerm is an Object Property as it has range instances of the class vsearch:ResultTerm. Moreover, it is worth mentioning that the definition of a vocabulary and its terms follows the formal way of an ontology specification using RDF and RDFS, like the definition of vsearch that can be acquired online by visiting the hosting webpage[24].

Furthermore, ontologies are employed to play the role of the data mediation layer by transforming domain concepts into vocabulary terms and defining the relationships between them. For example, in the e-Freight project's data mediation solution presented in [84], in the scope of this PhD, the biggest challenge that we had to address was the abundance of different data models used by the various stakeholders to exchange information about common concepts. Most of the organisations in the logistics market have developed their proprietary schema to model the same domain. Therefore, a company must adapt to a large number of different standardised message formats when conducting the appropriate transport logistics arrangements. Some of these formats may be custom developments by a single business or by a business cluster, others developed by official standardisation bodies (and often multiple standardisation bodies). This challenge was the main objective of the mediation layer that we provided through our partnership in the e-Freight project by creating the e-Freight ontology as presented in [31]. The ontology was the responsible module in a larger system, that would provide terms to represent all the needed concepts in the various messages that are exchanged among the network of stakeholders, i.e. port

[24] http://vocab.sti2.at/vsearch

authorities, shipping companies, logistic centres, etc.

Building and publishing a vocabulary

Designing a vocabulary is similar to the process followed to model a domain and design a solution for a given problem. Database schema design process could also be considered as a similar engineering task, although these two schemas are targeting to fulfil different assumptions, i.e. the open world assumption for the vocabularies ecosystem and the closed world assumption for the databases. One of the most important principles to follow, while designing a vocabulary is to try to reuse existing vocabularies, though without violating the semantics of the vocabulary terms as those are defined in the frame of the existing vocabulary. There are a few proposed workflows regarding the ontology reuse as presented in [74] by Simperl, which studies them and presents the common denominator among them. The study focused on 11 different approaches and the steps include ontology assessment, integration, translation between representation formats, extraction of structured data from text and customisation. Although the steps are very similar the approach in each case varies.

Apart from the important step of understanding how to effectively reuse existing vocabularies into new domains and problems, it is also crucial to enable the discovery of vocabularies by the prospective vocabulary engineer. In this respect, the work presented in [23] aims to shed light on this concern and defines the *Ontology library system.* According to the authors *"an Ontology library system is a library system that offers various functions for managing, adapting and standardising groups of ontologies. It should fulfil the needs for re-use of ontologies."*. In this respect, the Linked Open Vocabulary directory, presented in details under Section 3.1.1, aims to play the role of a centralised directory that the vocabulary engineer can use to search for existing vocabularies and terms.

In the process of publishing a vocabulary on the Web there are a few steps to follow and ensure that the vocabulary is introduced with the appropriate information, which will make it self-explanatory and reusable. The various parts involved in the preparation workflow are summarised below.

Host. The vocabulary needs to be hosted on some server under a publicly accessible URL. This server instance contains the physical files of the ontology and includes a presentation page, which helps the users to understand the various parts of it and the relationships between the classes and properties. Furthermore, a user should be able to download the source RDF or Turtle file from the same endpoint. The "http://vocab.sti2.at" subdomain of STI Innsbruck is such an example. It is used to host a few vocabularies developed within the institute. The webpage runs on Neologism [5], which materialised the vocabulary management system functionality that is needed to publish a vocabulary.

Persistent URL. The persistent URL is an optional feature of a newborn vocabulary, although not crucial for ensuring the success of it, it could support it. By the term persistent URL, we refer to a URL that remains the same across the time dimension and decouples the hosting of the vocabulary from the reference to it by other vocabularies. For many years, purl.org [50] plays this role. PURL stands for Persistent Uniform Resource Locator it easy for engineers to create persistent links to their vocabularies. If the author of the vocab decides to move the vocabulary under another host (e.g. the affiliation is renamed or the domain changes for any possible reason), then having a URL that does not change is really important for the reusability of it. Taking in consideration the following scenario will help to understand the importance of this step. When an author builds a vocabulary, she creates an RDF file that describes the vocabulary by referring to existing vocabulary namespaces. In case one of the referenced namespaces, assume it is called vocabA, goes out of service, then the new vocabulary will be pointing to an obsolete endpoint. However, the vocabA has been moved to a different domain. The options are only two, either the vocabA will need to notify every vocabulary author referring to it and request to update their references or the authors need to find it out on their own, probably spontaneously. However, if at the first place the author of vocabA had registered a persistent url under purl.org, then the domain change would only require an update at the purl.org website regarding the redirecting URL. Apart from that update there is no need to change anything else in the Web of Data ecosystem.

In this respect, studies [2] and fundamental paradigms [6] have been published to stretch the importance of building persistent URIs for the published documents (in the broader sense of hypermedia) on the Web. We could summarise all the above analysis by quoting Tim Berners-Lee from the article about persistent URIs [6]: "*A cool URI is one which does not change.*".

Vocabularies directory. An important role in the discovery of vocabularies is carried out by the Linked Open Vocabularies (LOV), Section 3.1.1, initiative. The LOV directory is not responsible for hosting the vocabularies, but only to play the role of the ontology library system as that presented before. All the versioning that we see is metadata. Also, the user can download the latest version of the vocabulary, but we should not get confused that LOV hosts it; the platform only stores the RDF file. Opening the downloaded RDF file, we realise that the namespace has the domain that the author hosts the vocabulary, or even better the persistent url.

Examining existing "healthy" vocabularies, we can realise the implementation of the above mentioned steps in the vocabulary publication process. For example, the "vrank" vocabulary, that is reused by the newly created "vsearch" in the context of this thesis, is hosted under the URL "http://vocab.sti2.at/vrank", which is the redirect target of the persistent URL "http://purl.org/voc/vrank". Finally, the "vrank" vocabulary is discoverable under the LOV directory with a profile page under "http://lov.okfn.org/dataset/lov/vocabs/vrank", which includes a brief description of its purpose and other relevant metadata.

Standardising a vocabulary

In parallel to the growth of the amount of vocabularies, there are also efforts led by the major search engines (i.e. Bing, Google, Yahoo!, Yandex) to standardise the vocabularies for the mostly used and searched domains. In this regard, the schema.org initiative was initiated by the major search engine providers[25] to provide a collection of vocabularies and a referral point for the rest of the ontology engineers to reuse terms from it. There are many discussions related to this initiative including criticism or constructive analyses, like the one presented in [61], which stretches a few design decisions that could be done in more explicit ways or differently. For example, quoting Patel-Schneider, he underlines that "it is unclear whether schema.org types and properties must be identified by URLs in schema.org, but all current schema.org types and properties are so identified."

The need of having standardised vocabularies emerged after the exponential growth of the vocabularies set at the early days when the impact of the vocabularies to the machine-understandable content direction was clear. In the scope of the presented research work, a *standardised vocabulary* as defined in Definition 8 is a vocabulary that is widely accepted both by developers and search engine providers.

Definition 8 (Standardised Vocabulary) ***Standardised Vocabulary*** *is a published vocabulary that has been accepted, recommended and recognised by the search engines and key stakeholders for the described domain.*

The equation about the effectiveness of the semantic annotations using semantic vocabularies includes two main factors apart from the content, i.e. the developer's efforts and the search engine provider's efforts. In case one of the two stakeholders is not aligned to the status quo, the effectiveness of the final result is very limited as far as the recognition of the vocabulary by the search engines is concerned. The search engines are using the semantic mark up (i.e. the semantic annotations) in order to better interpret the website content and be able to present it a more intuitive way to the user already at the search results, e.g. by showing a review score indicator for places, etc. Furthermore, when the search engine index of the crawled websites has better understanding of the crawled content then the probability of providing a more accurate and relevant set of results to the user is higher than it used to be in the old days that the crawl index was relying on statistical metrics and string similarity of

[25] Release timeline of schema.org: http://schema.org/docs/releases.html

the content and the keywords.

Another effort to provide a standardised vocabulary for a specific domain by the leading standardisation organisation GS1[26] in the world of business as fas as the communication formats, and language is concerned. The initiative is called GS1 Web Vocabulary and aims to provide a vocabulary for trading goods in order to achieve a better search experience for the users, greater visibility of the products in online searches and explicit description of product information in a unified way. Browsing through the various classes, or vocabulary terms as they are called throughout this manuscript, are sufficient to describe places, products, food products, clothing products and the related organisations and brands. All the described classes are presented in a layout that highlights the equivalence of that class with a term in schema.org, making the GS1 Web Vocabulary the first external extension of schema.org[27]. Another important contribution of this vocabulary to the Web sphere is the standardisation of the various enumerations about various types. The list of types[28] provides for each type code all the possible values and suggests that enumeration to be the standard across the various business stakeholders. For example, the *Diet Type Code*, which is a closed list of values can be different among the various airlines or the catering businesses. However, this vocabulary introduces a canonical enumeration that could be followed by all the stakeholders in order to smoothen and lower the walls in the partnerships and the exchange of information over the Web. Exactly this aspect is considered a very important outcome of the presented PhD approach in the scope of transforming websites to APIs. The next section discusses the information exchange aspect and the role of a vocabulary or ontology to it.

2.3 Semantic annotations as API

Exchanging information between various stakeholders requires a communication contract between the two ends in order to know what to expect and what needs to be pro-

[26] GS1 Web Vocabulary: http://www.gs1.org/voc/

[27] GS1-schema.org: http://blog.schema.org/2016/02/gs1-milestone-first-schemaorg-external.html

[28] List of GS1 Web Vocabulary type codes: http://www.gs1.org/voc/?show=typecodes

vided by the two ends respectively. This scheme applies to both business-to-business and consumer-to-business communication threads, or in other words provider-to-provider and user-to-provider in the scope of the Web. For example, a user would like to consume programmatically information from a specific provider (like a search engine, social network, etc.), but also a provider would like to programmatically build a communication bus with another Web provider (like an e-commerce search engine with a retail e-shop). Traditionally, the above described requirements are fullfiled by consuming the corresponding provider's API. Implementing the communication with a provider's API is in most cases an easy process, but still cumbersome as a few things need to be addressed, like the establishment of an authenticated communication session, possible limits in the number of the requests, etc. On the other side, the owner of an API has invested significant resources in order to build it and maintain it.

The need of providers to develop APIs in case they need to be interconnected with the other providers or users stems from the nature of the Web content presented via the webpages. Web content is in principle unstructured or structured but not machine interpretable and readable. Therefore, webpages are meant to be read by humans, the users of the Web, and not by machines. Addressing this problem led to the paradigm of the Web Application Programming Interfaces (Web API), which are designed as a set of endpoints by the owner of the service. This set of endpoints cover the information that the partners need to have access to in a machine understandable way. Another approach to address the main reason behind the existence of the API paradigm, would be the enrichment of the websites with structured and machine readable data. In this direction, the semantic annotations came into the Web surface. As it was described in Definition 5, semantic annotations are metadata about the presented information of a webpage. Therefore, the implementation of annotations on a webpage does not change the information that is presented to the user, while makes the content understandable to any programmatic usage via a Web agent or any other machine based solution.

The existence of semantic annotations in a Web document allows a search engine to better interpret the content in order to support more sophisticated questions to be answered, but also allows the development of a data store of information in a way that the data can be leveraged to many applications. The search engine providers do not request any API development from the various websites, but only parse and extract the annotated data in order to enrich their data repositories. Therefore, the requirement of the various Web based service providers to have access to an API in order to consume the data of their various partners could be simplified by using a semantic annotations parsing mechanism. In this way the annotations play the role of an interface to the unterlying data without the need of developing any additional API.

In addition to the above mentioned scenario, the semantic annotations' consumption becomes even more accessible and viable when the used model is part of an open standard that is widely used in the given domain. In this direction, the schema.org set of vocabularies can be considered as a standardised vocabulary according to Definition 8. A Web based business provider would need to develop only once the interpretation of the common vocabulary, which all the stakeholders would use to describe the publicly shared data. On the other hand, any secure communication threads, including transactional actions, cannot be completed with the above described paradigm, as they would require a secure endpoint.

2.4 Summary

This chapter walked the reader through the evolution of data on the Web starting from the very first publication of information on webpages. The spontaneous introduction of the tagging system in the Web ecosystem earlier than the Semantic Web vision, proves the emerging need of knowledge management within a chaotic data space like the Web. Comparing the initial stage of the Web data with the Linked Open Data paradigm, it is clear that the knowledge representation has moved towards more structured data patterns in order to be disseminated more effectively and to be consumed both by prospective users and agents without any external additional assistance. The demand of the current spectrum of applications on the Web to the Web data space is not only increasing in volume and velocity but also in quality aspects. Structured data has the potential to become machine-interpretable. Machine-understandable information empowers both users and organisations to accomplish their objectives more effectively in a day to day basis by ensuring an accurate interpretation of the information and direct data availability by providing the data points directly on the websites with interweaved meaning together with the markup of the HTML page source.

This demand of giving meaning to the content is explored in Section 2.3, which analyses the various technologies of the Semantic Web layer that facilitate the exchange of machine-understandable data. In the core of the Semantic Web technologies is the ontology concept, which has been briefly discussed in this chapter. The usage of ontologies is not only about a better online presence, but also about having a better connectivity among the various services and stakeholders on the Web. As presented in Section 2.3, there is a need to exchange data in a machine-understandable way, that can be leveraged to build partnerships with enhanced time regarding the ready-for-production curve, easier maintenance and more discoverable business data endpoints. In this respect the generation of a vocabulary based on existing vocabularies that can be used to represent and expose data will facilitate the transformation of already presented hypertext content into machine-understandable content that could also be used by prospective partners to build communication channels for data exchange and extraction.

For example, an application like the one developed in the scope of this thesis, namely the Accessible Vienna app shown in Figure 2-4, could benefit by aggregating data sources that provide explicit semantics about the transferred data. The effort to integrate the various sources would be easier as the data itself would explicitly define what it refers to, without the need to study any respective documentation or to go through a communication thread of questions to the data provider. In such a scenario, the underlying system would be enabled to automatically or semi-automatically integrate more data sources, as the latter would follow some existing and publicly shared vocabularies. At this point, it is suitable to remind to the reader that throughout this manuscript the terms ontology and vocabulary can be interchangeably used as it was already stretched out in Section 2.2.2.

Moving forward towards the designed approach of this thesis, the next section elaborates on existing tools that would help to bring the Semantic Web layer closer to the data space. Specifically, it focuses on existing approaches regarding the vocabulary exploration, ranking, the development of semantic annotations and the discovery of vocabulary terms for a given document. Therefore, Chapter 3 presents the related work to the thesis approach and emphasises on the gaps that need to be addressed.

Chapter 3

Related work and State of the art

Harnessing the power of the Web of Data

Having analysed in the previous chapter the various dimensions of the Web of Data, including the various types in which we could classify the data sources (Section 2.1), and the building blocks that have been introduced by the Semantic Web endeavours (Section 2.2), it is the high time to move one step further and examine how existing approaches function on top of the Web data in order to produce insights regarding the published structured data. The below described existing initiatives are related to the research space around the proposed approach of this manuscript. The related work space of the presented approach is mainly related to work relevant to vocabularies recommendation, ranking of vocabularies, usage of the LOV directory and exploitation of the LOD data for vocabulary terms discovery.

The workflow of Figure 3-1 demonstrates the basic steps required to define the

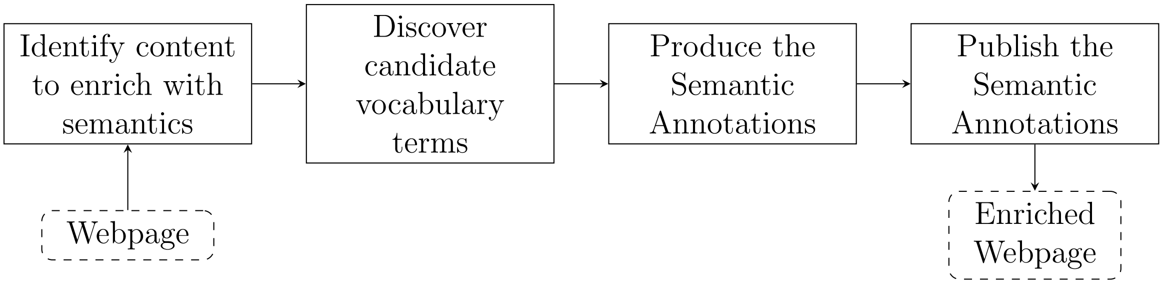

Identify content to enrich with semantics
Discover candidate vocabulary terms
Produce the Semantic Annotations
Publish the Semantic Annotations
Webpage
Enriched Webpage

Figure 3-1: The semantic annotations generation workflow without the usage of a domain ontology.

semantics of webpage content and publish the structured data of the webpage in the form of semantic annotations. The initial step is to identify what is the important content and entities that the webpage presents. For example, in a product description page there are information about the various specifications of the product, like dimensions, weight, images, price, manufacturer and actions that could be performed to interact with the described webpage entity like a purchase hyperlink. After having defined all the information bits that are considered relevant for the definition of the presented content, the next step is to find the appropriate terms that will encapsulate and transfer the semantics that we would like to add to the published webpage. As it has already been defined in Section 2.2.2, the terms that specify the semantics are members of a set of terms, the vocabulary. The vocabulary apart from defining individual terms, it also defines the relationships among them and potentially with external vocabulary terms. The selection of the vocabulary terms is one of the most difficult steps as it involves a few sub steps and iterations apart from the prerequisite of understanding the ecosystem of vocabularies in general. Once the vocabulary engineer has identified a shortlist of terms that could be used to define the information, then it is important to examine if those terms are sufficient in practice by trying to apply them on the content. Furthermore, it is also possible to be difficult to combine the terms from various vocabularies together. The upcoming Section 3.1 discusses the vocabulary exploration in detail, while Section 3.2 dives deeper to a special aspect of the vocabulary exploration, the ranking of the existing vocabularies. Naturally, any question that has more than one answers, the answers need to be evaluated and assigned a score in order to be eligible for sorting. Finally, the last step refers to the implementation of the annotations using one of the formats that were described in Section 2.2.1.

This chapter functions as the state of the art analysis in the scope of semantic annotations generation, which is related to the proposed approach. It starts with the description of vocabulary exploration tools (e.g. LOV in Section 3.1.1 and vocab.cc in Section 3.1.3), which are also used within the proposed framework. Furthermore, in the context of the vocabulary exploration, the following one, Section 3.2 presents the related work in the field of ranking vocabularies and vocabulary terms, which is one of the basic aspects at the core of the presented approach. Section 3.4.1 demonstrates the manual effort needed to produce semantic annotations by following all the steps in Figure 3-1, as part of a survey that was run in the context of the conducted research in order to identify the difficulty and the obstacles in the process with quantitative and qualitative metrics.

Similarly to the motivation of this thesis, the open issue of assisting the semantic annotations development has been a topic for research for many groups. Section 3.4 discusses the facilitation of the semantic annotations development by describing a set development tools that support the development of annotations either via providing an enhanced editor or via testing platforms. In addition, a few approaches related to the semi-automatic or automatic annotations development are examined and presented within the second half of the section.

Last but not least, another important dimension, that is presented in the scope of the related work in Section 3.5, refers to the collaboration of the various contributors for the development of a vocabulary via versioning control systems or other platforms.

3.1 Vocabulary exploration

The discovery of the relevant vocabulary terms to annotate a web document requires some basic prerequisites in order to be effective. In this scope, a few approaches have been developed to facilitate the discovery of vocabularies. The broadly used search engines could be very good services to find information about any topic that a user could be seeking information for. However, as far as discovering vocabularies is concerned, it could become cumbersome to browse through results and locate those that are describing vocabularies. Furthermore, the crawling algorithms of the search engines and their ranking score formulas do not seem to be relevant for the various vocabulary description pages, as they are mostly technical documents not optimised for the Web content information indices. Therefore, solutions like the Linked Open Vocabularies directory, described in Section 3.1.1, are considered crucial for the search and exploration of the vocabulary space. In addition, other approaches aim to support the exploration of Linked Data and the usage of the vocabularies in the Linked Data cloud.

3.1.1 Linked Open Vocabularies

In the chaotic landscape of vocabularies scattered in the Web sphere, the Linked Open Vocabularies (LOV) [91] initiative aims to put some order by providing a comprehensive directory of all the existing vocabularies. Main purpose of the platform is to bring together all the vocabularies, document them and store the documentation in a single repository. This approach enables the users of the platform to discover vocabularies and get information in a uniform way about the described vocabularies.

The vocabulary space of the Semantic Web includes more than 500 vocabularies according to the LOV repository. This space has been populated by domain experts and researchers to facilitate the interpretation and exchange of information in the Web of Data. The abundance of vocabularies and terms available in the LOV space, on one hand aims to cover the major knowledge management needs but on the other hand it could be cumbersome for a non-expert or even a vocabulary expert to find the correct way through the collection. A visualisation of the vocabulary data in the form of a bubble chart, as shown in Figure 3-2, welcomes the user at the front door of the website[1], which gives an indication of the amount of the listed vocabularies and their popularity. The diameter of the circles indicate the popularity of the vocabulary within the LOV data sphere.

LOV has a fundamental role in lowering the barriers of the Semantic Web adoption in the plateau of productivity by providing a curated directory of vocabularies and search functionality on top of the curated data. Each vocabulary is accompanied with a profile page[2] that provides useful metadata about the vocabulary itself, e.g. the namespace, the number of classes and properties of the latest version, the number of

[1] LOV is accessible at: http://lov.okfn.org/dataset/lov.

[2] For example, the schema.org profile at the LOV directory is available under the URL: http://lov.okfn.org/dataset/lov/vocabs/schema

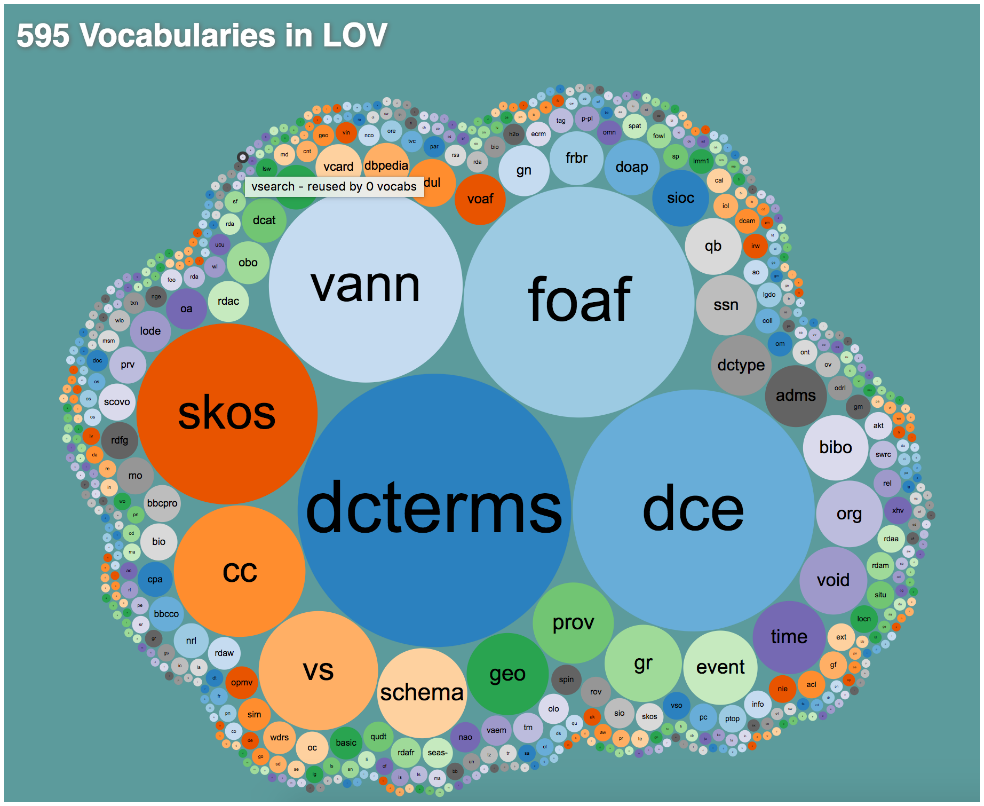


Figure 3-2: LOV space snapshot from March 2017 (screenshot captured from the LOV website). The size of the circles reflect the number of incoming links of the vocabulary compared to the rest. The dcterms vocabulary is the most interlinked with 488 incoming edges. The less popular ones without any incoming link are placed at the peripherals of the shape with the minimum radius. For example, the vsearch vocabulary, which was developed in the scope of the presented research work, is highlighted and the label shows that it has not been reused yet by any other vocabulary.

incoming and outgoing links, the versions history, the authors and the raw vocabulary schema in N3 notation. In addition, a number of discovery interfaces have been implemented including a search for the vocabularies and the terms in order to find the most relevant resources for a given keyword; a SPARQL endpoint to query the data repository; and a JSON based REST API that facilitates the integration of the search functionality with external applications, like the approach that we present in this paper.

Furthermore, the LOV creators introduced a new search capability, i.e. the agent search. This enables the user to search about specific contributors to vocabularies. Interestingly enough, a statistical report published in [90] about the number of queries and the distribution of them across the various search types, shows that 74% of the total searches refer to agent searches. The figures refer to a six months window in 2015 and shows that the LOV platform served 1.4 million queries in total. Furthermore, 92% of the queries with a keyword were made for terms, while only 39% of the total number of term searches are using keywords and not the various filters provided on the terms search page.

3.1.2 Schema.org

Similarly to the main motivation behind the LOV creation, another consortium began an initiative in 2011 to provide a solution to the overwhelming amount of vocabularies that have been published in the Web sphere. The consortium included the major search engine providers, i.e. Google, Microsoft, Yahoo and Yandex by that time, while the product of this cooperation is known as the schema.org which can be accessed at the respective webpage. This approach is in the opposite direction from the one that the LOV directory is heading to, but equally important and well accepted by the Web community. Schema.org is actually a set of various vocabularies interlinked. This vocabulary, or this set of vocabularies, aims to model the major domains that would benefit from structured data, like local businesses, e-commerce, etc.

According to the organisation page of schema.org, the current version consists of 583 Types, 846 Properties, and 114 Enumeration values[3]. Most of the types are classes related to a few major domains, like Product, Place, etc. A visualisation of the schema.org schemas is depicted in Figure 3-3 by using the WebVOWL tool[4]. As shown in the graph of the figure, there are a few central classes that have a lot of properties connected to them or other classes that extend the former. The various groups of classes are slightly connected with each other, but still the whole set of schemas is designed with some interlinking in place.

The schema.org maintenance is one of the very important aspects that need to be addressed in order to ensure the longevity of the initiative. In a respective section[5] of the website this topic is discussed, by explaining the versioning, the extension mechanism and other management related topics. One of the important characteristics of schema.org is the standard quality of the models that comprise it. The community around schema.org organises itself mostly via the GitHub issue tracking system[6].

Finally, the presentation of the various classes and their properties is following a very simple and understandable tabular presentation. A very helpful asset of the website in contrast to the many other vocabulary websites is that most of the defined classes include examples in Microdata, JSON-LD and RDFa in order to facilitate the application of the vocabulary on webpages. This is one important step towards the uptake of the vocabularies by the development community.

3.1.3 Vocab.cc

The vocab.cc service [76] provides metrics about the Linked Open Data (LOD) usage patterns (LOD is explained in Section 2.1.2) of the various vocabulary terms that appear in the LOD cloud. Vocab.cc has analysed the crawled dataset of the Billion Triples Challenge Dataset (BTCD) [37], and applied two frequency measures to the identified class and property URIs, i.e. a) overall frequency of URI use in the BTCD;

[3] Schema.org overview page: http://schema.org/docs/schemas.html

[4] WebVOWL visualisation of schema.org: http://vowl.visualdataweb.org/webvowl/#iri=http://www.w3.org/2012/pyRdfa/extract?uri=http%3A%2F%2Fschema.org%2Fdocs%2Fschema_org_rdfa.html&format=n3

[5] Schema.org maintenance workflow details: http://schema.org/docs/howwework.html

[6] GitHub issue tracker for schema.org: https://github.com/schemaorg/schemaorg/issues

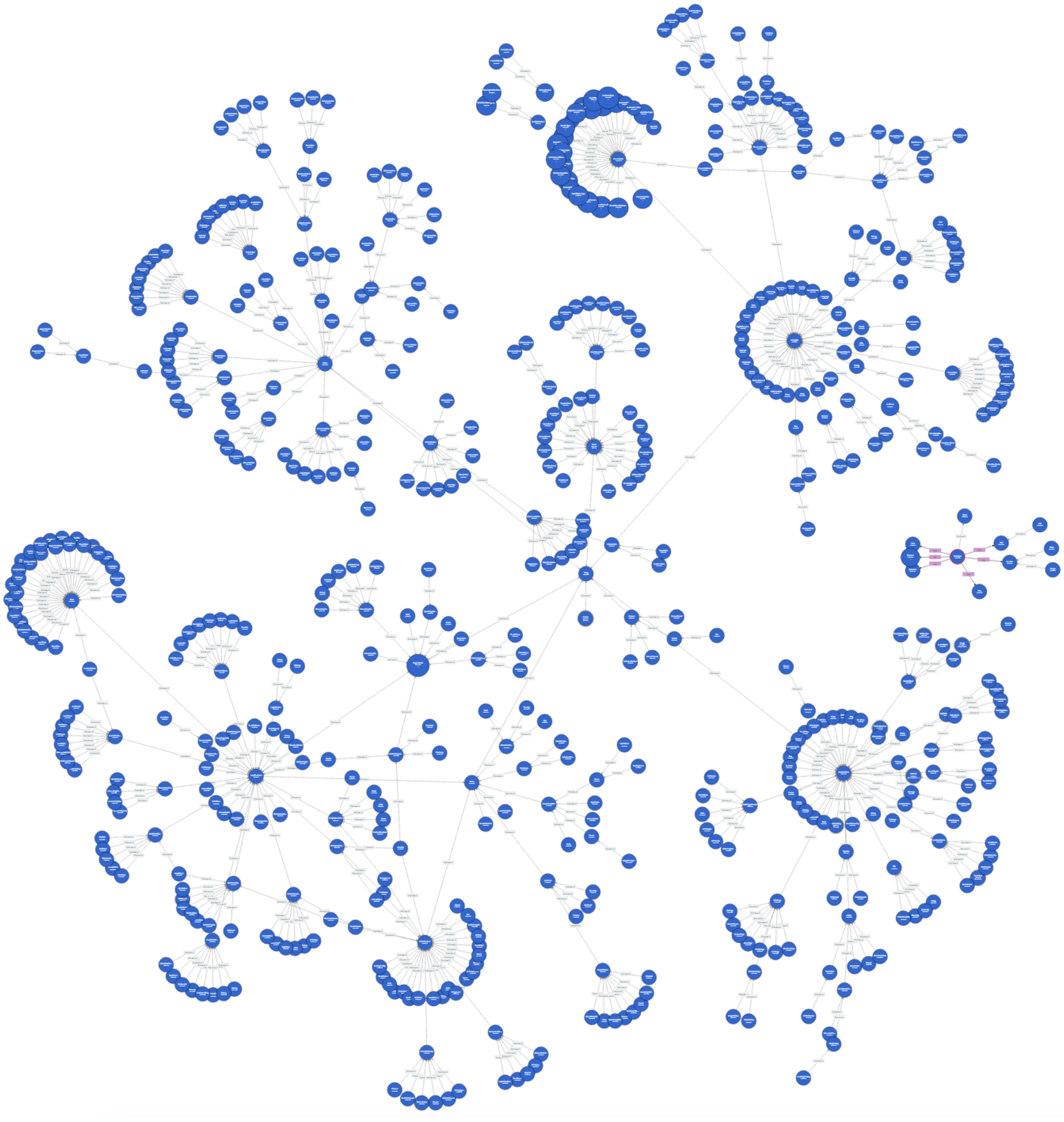


Figure 3-3: Schema.org visualisation using the WebVOWL tool.

Query keyword	Vocab.cc terms	LOV terms
recipe	-	49
ingredient	2 (from dbpedia.org, dbtropes.org)	47
museum	11 (from schema.org, dbpedia.org, dbtropes.org, sw.deri.org, ontologydesignpatterns.org.it)	396
hotel	5 (from schema.org, dbpedia.org, dbtropes.org)	31

Table 3.1: Comparison of results in vocab.cc and LOV for a few keywords by comparing the number of terms that vocab.cc provides as possible results and the number of matching terms returned with the LOV search.

and b) document frequency of URI use (how many different documents refer to them).

One weak point about the vocab.cc service is the dataset underneath, which is not dynamic but the *Billion Triples Challenge 2012 Dataset*[7], which has not been updated since 2012, while it could at least be using the newest one [43]. The BTCD dataset reflects the data collected during the crawling that took place in the scope of the challenge. Taking in consideration the velocity and volume rate of data publication on the Web, it seems to hinder the reusability of the service outside of prototyping due to the limitations that it introduces by being based on a old dataset. For example searching in vocab.cc for the terms recipe, ingredient, museum and hotel we realise that the results are limited as shown in Table 3.1. For *recipe*, the result set is empty, meaning that there is not any vocabulary term that could match the query keyword *recipe* and has been used in the crawled dataset of BTCD. On the other hand, as we can see at the third column of the same table, the LOV directory returns 49 matches for the keyword *recipe*. Similarly, we realise that all of the example keywords return 0 to 5 different term URI's host names in contrast to the many more results returned by the LOV service, which varies from 31 to 396 different vocabulary terms respectively.

Apart from the vocab.cc stats, another source of related metrics is the LODStats, presented in Section 3.1.4, which aims to provide a comprehensive picture of the current state of the Web of Data.

[7] http://km.aifb.kit.edu/projects/btc-2012/

3.1.4 LODStats

Reusing existing datasets is a major topic in the LOD space, in a similar way to the vocabulary reuse that is is thoroughly discussed through this thesis. Thus, it is very interesting to understand the dimensions that we could evaluate datasets and apply a scoring range on them. We could wonder, why should one evaluate the existing datasets in order to reuse them? The main issue is that we are not living in a perfect world, therefore all the existing approaches and solutions potentially have some strong points but also carry some weak points that should be accounted before making any commitment and consuming a source. As it is mentioned by Auer et al. in [4], it is important to know the structure, coverage and coherence of the data in order to choose which one fulfils the needs of a given use case. The structure of a dataset refers mainly to the vocabulary and properties usage by the included data records, while the coverage is the level of properties usage including the range for those that are interesting to study from a quantitative point of view. For example, a dataset about the post codes of a country is interesting to check what value range it has in order to decide on using it or not. In case the former post code dataset is limited to only a small area then probably we would like to seek for a better source of post codes rather than limiting our approach by selecting the wrong dataset.

In this frame, LODStats materialises a set of 32 different statistical metrics that are used to evaluate any given dataset. Furthermore, as it is proved by its authors it is much faster compared to similar approaches due to the stream based computation that it uses. The stream reasoning feature enables its processes to handle millions of triples and scale up better than other approaches. The datasets are stretched against various dimensions like quality analysis, coverage analysis, privacy analysis and link target identification. In our approach, a subset of the defined criteria are recognised as relevant and have been incorporated in our scoring algorithms that are presented later in Chapter 5.

The LODStats directory includes 2471[8] vocabularies, which is much more than the 562 vocabularies of the LOV directory. The difference is due to the numerous

[8] At the time of the dissertation writing, 19.08.2016.

vocabularies that are met in the LODStats repository, which refer to very specific domains and use cases. For example the the *geodati* vocabulary[9] is used in 138 datasets, but all of them are part of the domain that hosts the ontology. Therefore, as this specific vocabulary is tailor made to the needs of its creator, it does not add any value to be shared via the LOV directory. Furthermore, there are vocabularies, like *http://www.systemone.at/2006/03/wikipedia*, which is a custom ontology to describe the Wikipedia snapshot of the respective timestamp in RDF. According to LODStats this RDF dump of 3.6 GB includes ca. 47 M triples and 2 M links. A property of this ontology appears to have more than 34 M occurrences and to be trending at the top 5 properties, although it is used only in this specific dataset, as shown in *http://stats.lod2.eu/properties/3116*. Therefore, the various statistics that appear in the LODStats need to be treated with care as they could be skewed due to outliers introduced by datasets similar to the two aforementioned cases.

3.2 Vocabulary ranking

An approach presented by Atemezing and Troncy in [3], examines the problem of vocabularies recommendation based on a ranking metric that has been developed by introducing the concept of Information Content (IC) to the context of LOV. In comparison to the methodology proposed in [3], we follow a more holistic approach by starting earlier in the funnel of searching for vocabulary terms to annotate a given web page content, while the IC approach aims only to rank the vocabularies. The IC approach aims to rank the vocabularies by evaluating the terms occurrence in comparison to the maximum term occurrence in the set of vocabularies and then leveraging the term rankings with a sum and a weight depending on the centrality of the vocabulary in the set. However, we still consider it relevant to the *LOVRank* method that is presented later in Chapter 4.

Furthermore, Butt et al. in [14], [13] aim to address the ontology ranking problem by introducing the DWRank algorithm. DWRank consists of two main scores the Hub

[9] http://www.territorio.provincia.tn.it/geodati/ontology/

score and the Authority score, which measure the centrality of the concepts within the ontology and the ontology authoritativeness (i.e. the importance of the ontology in the ontologies space), respectively. DWRank and the proposed approach share the same perspective of ranking the concepts defined within the ontologies in order to find the best match to a keyword search. One of the main differences in comparison to the proposed approach is related to the consideration of the LOD cloud as an input to the algorithm, which is included in our approach as presented later.

Discovering vocabularies can be assisted via many different directions apart from the ranking of vocabularies and vocabulary terms. Schaible et al. in [69], aim to support the ontology engineer with the *LOVER* framework by providing a methodology that guides the creation of semantic annotations (Linked Data) through the best practices for modelling new entities [38]. The *LOVER* approach is mainly based on *Swoogle*[10] and the *SchemEX* index and consists of an iterative process, where each iteration cycle finishes with the definition of one or several mappings of data to vocabulary terms. Furthermore, the vocabulary reuse is studied in [70] by presenting various approaches and one of the many extracted insights is the fact that using popular terms from popular vocabularies is preferred over using mainly one domain specific vocabulary that covers the needs of the given data. This insight is taken in consideration in our approach and reflected in the formulas presented later.

TermPicker presented in [71] experiments towards the direction of suggesting types and properties from vocabularies that other LOD providers have combined together with the one the engineer has used to model the given part. To achieve that, the authors introduce the schema-level patterns (SLPs), that represent the connection between two sets of RDF types (vocabulary classes) via a set of properties.

Ellefi et al. in [25] propose an approach to recommend datasets to a given non-linked dataset. The aim of the recommendation framework is to provide the user with an ordered list of datasets that are potential candidates for interlinking with the given input dataset. They base the interlinking on building a profile graph that provides information about the relationship between a document and a topic by following a

[10] http://swoogle.umbc.edu/

Relationship	**Example**
Metadata	Using dct:title
Import	Using owl:imports
Specialisation	Using rdfs:subClassOf, rdfs:subPropertyOf
Generalisation	Using skos:narrowMatch
Extension	Using owl:inverseOf, rdfs:domain
Equivalence	Using owl:equivalentClass, owl:equivalentProperty
Disjunction	Using owl:disjointWith

Table 3.2: Inter-vocabulary relationships and examples that help to identify them in the definition of a vocabulary.

topic modelling process.

Finally, the LOV directory, which was introduced earlier in this chapter, provides vocabulary and terms search functionality, which is leveraging an internal ranking methodology. This methodology as described in [90] aims to promote the reuse of vocabularies that are widely already used. The LOV scoring methodology incorporates a popularity metric that reflects the terms usage in respect to the frequency and the number of datasets using it. In addition, the scoring equation includes the term frequency inverse document frequency (tf-idf) to account the relevance of the query when assigning a weight to a particular vocabulary for a given query term. Furthermore, in the latest publication about LOV ([90]) the contributors have classified the relationships between vocabularies (inter-vocabulary) that are taken into consideration in the following groups: *Metadata*, *Import*, *Specialization*, *Generalization*, *Extension*, *Equivalence*, *Disjunction*. All of the relationship types refer to the various classes and terms that are defined within the vocabularies and how they are reused by other vocabularies in the definition. For example, a vocabulary could be extending a vocabulary by using a class of the latter as the domain for a property of the former. Table 3.2 shows the taxonomy of relationships together with examples that would help the reader to identify such inter-vocabulary relationships.

An overview of the abovementioned approaches is summarised in Table 3.3.

Approach	**Methodology**
Atemizing et al.	Information Content (IC)
DWRank	Centrality, Authoritativeness
TermPicker	Schema Level Patterns (SLP), LOD
LOV	Popularity metric, LOD
LOVER	LOD, Swoogle, SchemEX

Table 3.3: Overview of the vocabulary recommendation related work.

3.3 Manual semantic annotations development

Seeking tangible figures regarding the process of discovering vocabulary terms for a given webpage by using the search means of the LOV repository, a survey was designed to gather data about the difficulty and success of engineers to work with vocabularies. In this respect, as it has been published in [80], the distributed survey aimed to measure the time and the results of the participants and use it as a reference point for the evaluation of the proposed approach. Therefore, the steps of the survey included a) the discovery of the appropriate vocabulary terms for a given webpage and b) the creation of the related JSON-LD snippet.

The participants were asked to provide a justification on their decision of the used terms over other candidate vocabulary terms and also to specify their main difficulties throughout the above two steps. In addition, they were asked to measure the time needed to complete the two tasks separately and they were given a relaxed timeframe of one week to complete the tasks. All of them were familiar to computer science topics, but without any similar experience with semantics. For the soundness of the survey presentation, the various metrics about the survey answers and the survey questions can be found under Appendix A. The demographics regarding the expertise of the participants is depicted in Figure 3-4, which shows that the survey's population is not aware of the semantic annotations, vocabulary exploration and related usage topics.

The steps of the assignment regarding the discovery of vocabulary terms that the participants were expected to complete are the following:

1. Browse through the given webpage and extract the important topics and key-

words.

2. Visit the Linked Open Vocabulary search engine and search for the extracted keywords.
3. For each searched keyword decide on the top candidate terms and choose one that could be used to describe the corresponding information bit in the webpage.
4. Note all the above vocabulary terms and provide them as answers to the assignment.

In order to introduce variety in the examined content types, the four following tasks were chosen in total and randomly each participant was assigned one of them. However, the distribution of the cases among the participants was uniform, as shown in Figure 3-7.

- *Article* from the NASA news[11] online feed.
- *Hotel* room webpage[12] from Austria.
- *Museum exhibition*[13] webpage of Louvre.
- *Recipe* webpage[14] for a pizza.

The four use cases listed above require a different set of vocabulary terms and belong to a different domain. Although the various use cases refer to different domains, there is a core set of terms that are shared among the webpages, that can be used to describe basic elements of a webpage, like an image, a title of an entity or a hyperlink to another resource. The news article use case refers to the discovery of evidence of water on planet Mars including pictures, address of the author and publication date among other details. The hotel webpage describes a hotel room offer mainly populated with prices, images and listing of the amenities. The museum use case refers

[11] http://www.nasa.gov/feature/jpl/nasas-curiosity-rover-team-confirms-ancient-lakes-on-mars

[12] http://www.mohr-life-resort.at/zimmer-und-preise/detail.html?rid=12

[13] http://www.louvre.fr/en/expositions/winged-victory-samothracerediscovering-masterpiece

[14] http://www.cookingchanneltv.com/recipes/debi-mazar-and-gabriele-corcos/margherita-pizza.print.html

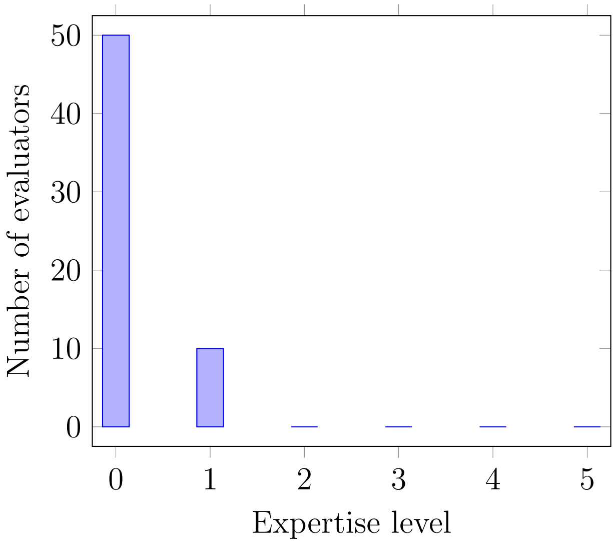


Figure 3-4: Expertise level distribution of the participants base.

to a museum exhibition page, which informs the visitors about the visiting period, the title of the exhibition, the admission fee and other related information. Finally, the cooking recipe use case is one of the most common semantic annotation examples as one of the major search engines has launched in the past a recipe specific search by leveraging the structured data of the ingredients to filters[15]. The recipe webpage shared in the survey describes all the steps, the ingredients, serving details and nutritional data of the recipe. The distribution of the evaluators to the use cases is almost uniform with an average of 16 evaluators per use case and a total of 64 evaluators with valid submissions (two more submissions were considered invalid as they were not meeting the expected standards); detailed percentages are shown in Figure 3-7. Furthermore, a detailed presentations of the use cases is analysed in Chapter 7 by presenting the webpages themselves together with labels that reflect the parts that can be annotated.

Identifying the keywords on a webpage that would be used to map vocabulary terms to is a tedious task and the time needed varies due to the complexity of each use case.Figure 3-5 demonstrates the correlation of the time needed with the various use cases by putting all the distribution histograms of the four use cases together

[15] Google recipe search: https://googleblog.blogspot.de/2011/02/slice-and-dice-your-recipe-search.html

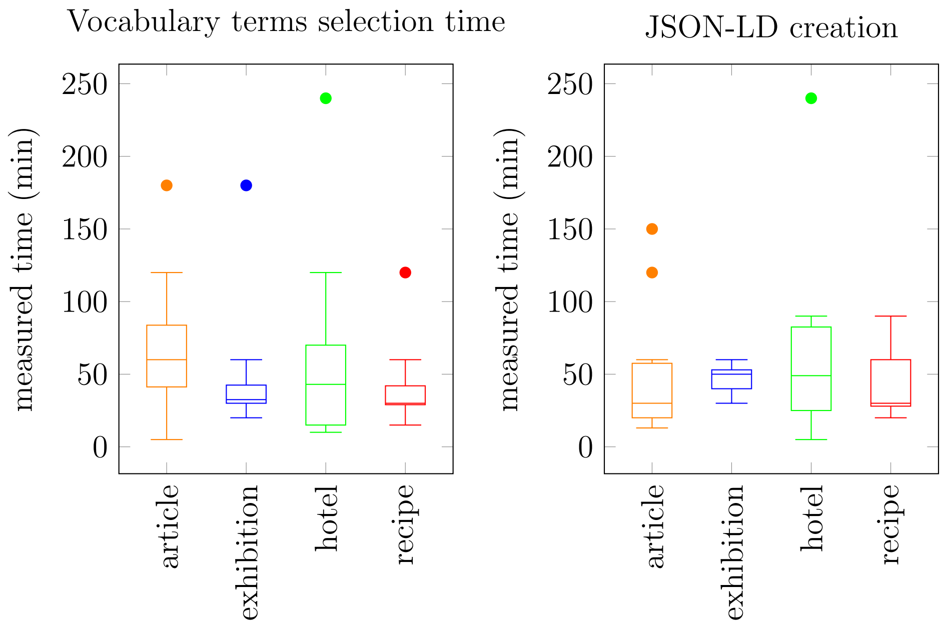


Figure 3-5: Distribution of the time needed by the evaluators both to select vocabulary terms and to build the JSON-LD snippet, grouped by the 4 experiments (published in [80]).

in the form of box plots. As it is depicted on the box plots, the article case holds the highest median value while the recipe case has the lowest measured time median together with the exhibition. Also, it is interesting that the article and the hotel cases are those with the most skewed distributions in the experiment, which gives an indication that the evaluators interpret differently the search results and that they searched for a different number of keywords.

The box plots diagrams allow to compare the various distribution diagrams of all the use cases at the same time. The boxes represent the range in which the 50% of the data points fall in, while the two horizontal lines (whiskers) above and below the main box part refer to the maximum and minimum values respectively. The dots depict the outliers in the dataset and the horizontal line in the box the median.

Analysing the term URIs that the evaluators proposed and used for the JSON-LD generation, gives indicators about the pitfalls that are hidden in the transformation of a webpage to an annotated data node. Furthermore, taking in consideration the

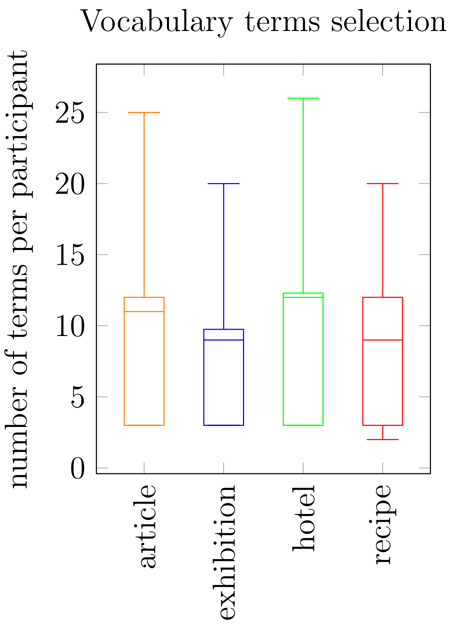


Figure 3-6: Distribution of the number of selected terms per participant for each use case (published in [80]).

reasons of their decisions we can realise that a few basic requirements should be met in order to make a vocabulary an option and potential solution for the vocabulary needs of the webpage development process.

Figure 3-6 shows that the median of the amount of selected terms across all the use cases is between 9 and 12, while are no outliers are observed, albeit the fact that the maximum values are roughly twice big compared to the 3^{rd} quartile (upper edge of the box). The hotel room page has the highest median and maximum number of terms, which can be justified by the fact that the content of the respective webpage includes easily recognisable entities, like the room rate per night, amenities, address, etc.

In total the participants proposed ca. 500 term URIs, while the used terms are 300. The difference is due to the fact, that they were asked to include in their selection both the terms used at the JSON-LD generation later and the alternative candidate terms. For the one third (33%) of the proposed term URIs, the namespace is the schema.org, while from the used ones the schema.org terms are 47%. This result

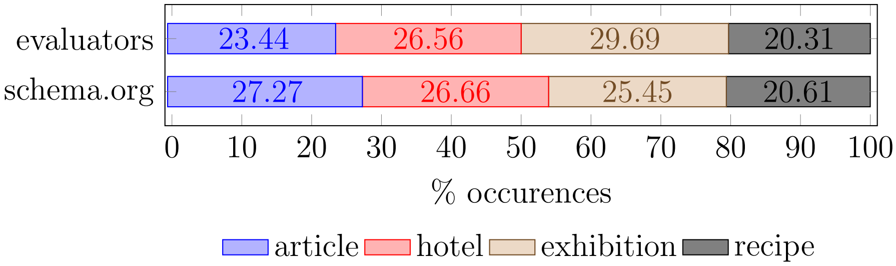


Figure 3-7: Distribution comparison of the number of evaluators per use case with the schema.org terms usage per use case (published in [80]).

shows a trend of the schema.org terms being in favour over the rest of the candidate terms selected. The participants justified their decision on the good documentation of the vocabulary, the lack of documentation of terms in the rest of the vocabularies or even vocabulary websites not accessible due to technical issues (like 404 pages). Observing the high usage of schema.org in the results, a question arose: *"Do any of the use cases yield more schema.org terms in the selection set?"*. Figure 3-7 helps to answer such a question by showing that the schema.org terms are following roughly the same distribution like the one of evaluators number per use case.

The title of the survey reflects the main focus of it, i.e. the manual selection of vocabulary terms for the given webpage. Additionally, the second assignment of the survey refers to the generation of JSON-LD snippets by utilising the vocabulary terms result set. According to Figure 3-5, this task has a less skewed distribution with the min and max values closer to the main box, while it seems to be taking 30 to 60 minutes in average across the four experiments. Combining the insights from the two tasks of the assignment, we have enough indications that the selection of vocabulary terms and the related decisions that need to be taken at that stage make the task more difficult than building the code snippet that will help to leverage the webpage to an annotated Web entity. The main detected issues were related to understanding the structure of the JSON-LD snippets and the syntax of it. The participants were given pointers to the examples of the JSON-LD playground[16], which helped them to understand how to apply the JSON-LD syntax to their use cases and the selected

[16] JSON-LD playground: http://json-ld.org/playground/

vocabulary terms.

3.4 Assisted semantic annotations development

Producing semantic annotations includes many steps until the deployment of those as metadata to the existing web content. The publication process depending on the infrastructure of the webpage is either completely manual (in the case of a static website) or less manual (in the case of using a Content Management System). However, in both cases there are many steps prior to the generation of the annotations that require significant development effort. The workflow diagram of Figure 3-1 demonstrates the various steps that are followed to design the semantic annotation mappings between the content and the vocabulary terms that specify the meaning of the presented information.

The ultimate target of the proposed approach is the facilitation of the semantic annotations development. Although, the core of the functionality is related to vocabulary ranking and exploration, this secetion aims to provide an overview of the various tools that assist the generation of annotations but also compare the various methods, which are designed to address the scalability issue of the semantics on the Web in autonomous or semi-autonomous ways. In this scope, Section 3.4.1 provides an overview of a set of tools that would empower the developer of semantic annotations, and Section 3.4.2 analyses the various existing methods that can be employed in order to annotate a bigger number of documents.

3.4.1 Semantic annotations development tools

This section provides a list of tools that aim to support in various ways the development of semantic annotations. All the tools described refer to the manual development of annotations and can be used to assist the creation of the appropriate syntax.

Structured Data Testing Tool by Google. Google being one of the major search engines, provides to the developers and website owners a simple to use testing tool[17] for structured data. The tool fetches a given URL and parses the underlying markup and source of the webpage in order to detect structured data.

Microformats generator. The authors of the microformats approach have developed a few online generators[18] of various microformat types (i.e. hCalendar, hCard, hReview), which can be used to produce in an easy way an HTML code snippet. This code snippet includes the typed in content by the user alongside with the syntax of the microformats semantic annotations. For example, the HTML code of Listing 2.1 has been produced by using this online tool.

JSON-LD Playground. JSON is considered to be one of the best modern data formats to exchange data on the Web after XML nowadays. Learning to produce and consume JSON objects is easy enough to make it the most popular format in the APIs world. JSON-LD objects are still valid JSON objects, while introducing some special members, e.g. @context, that allow the description of the type of the presented objects as shown in Section 2.2.1. For this reason, the creators of JSON-LD provide an editor assistant[19] or as they call it "playground", which can be used to validate a produced JSON-LD snippet or to study existing examples and how those are interpreted by the various parsers.

Schema.org generators. A significant amount of generators exist online that provide templates to build schema.org items in the format of microdata. Those generators consist of various templates following the schema.org attributes for a various item types (classes) that the development team of the generator has decided. In most of the cases, those generators are websites with various templates in separate webpages following the hierarchies of the schema.org set of vocabularies. Some of the websites that are considered to fall in this category include: http://schema-creator.org/,

[17] https://developers.google.com/structured-data/testing-tool/

[18] http://microformats.org/wiki/code-tools

[19] http://json-ld.org/playground/

http://www.microdatagenerator.com/, http://tools.seochat.com.

RDFa Content Editor. Furthermore, various related efforts in the field of Linked Data inclusion have been proposed and developed, e.g. the RDFaCE (RDFa Content Editor) presented in [45]. RDFaCE is a what-you-see-is-what-you-mean (WYSIWYM) content editor with various external services integrated to facilitate the generation of RDFa for the given webpage. It is based on the TinyMCE[20] editor, which enabled the user to edit HTML in a visual way without having to deal with the source code.

Less sophisticated RDFa generators can be found online that target a specific domain or schema. For example, in the scope of the GoodRelations vocabulary as presented in [40], the owner organisation has developer a rich snippet generator[21] that produces RDFa for a few entity types (schema classes), like a company profile, a shop description and individual products or services. The generator produces the HTML code needed to be added and in addition the documentation provides simple steps to be followed for the update of the webpage that will be enriched.

The process for the inclusion of Semantic Web markup would be to redesign and redeploy the websites by adding the needed metadata by adopting a Semantic Web format and a relevant vocabulary. The format is responsible for the technical publishing of the metadata on the webpage, and the vocabulary provides the terms and relationships that can be used to describe the content. The main search engines, including Bing, Google, Yahoo!, announced in 2011 a joint effort to create and support a common set of schemas for structured data markup on web pages, i.e. schema.org [32]. A lot of vocabularies link to schema.org and vice versa, as presented in the Linked Open Vocabularies, in order to achieve a rich representation of information.

Table 3.4 provides a brief comparison in two main directions (output format and used vocabularies) of the various tools that have been considered to sufficient to facilitate the process of semantic annotations generation by end-users. A more thorough

[20] https://www.tinymce.com/

[21] http://www.ebusiness-unibw.org/tools/grsnippetgen/

Approach	Format					Vocabulary, Knowledge Base
	Microformats	Microdata	RDFa	JSON-LD	HTML	
Google Structured Data Testing Tool	✓	✓	✓	✓	✓	schema.org
Microformats generator	✓					hCard, hCalendar, hReview
JSON-LD Playground				✓		any
Schema.org generators		✓				schema.org
RDFaCE			✓			any

Table 3.4: Comparison of the various tools that facilitate the generation of the Semantic Annotations either by providing templates, by validating or by providing step by step guidance. The comparison dimensions are related to the supported format and vocabularies.

evaluation in this area is presented in [44] and [41], where it is proved that no matter which approach is considered to be the most user friendly or effective they still require significant manual effort. Furthermore, it is important to note that the short list of tools described in this section refers to online available tools that are considered easy to be found by end-users or domain-experts that aim to develop semantic annotations.

3.4.2 Semi-automatic semantic annotations development

Manual annotation is considered an expensive process as it requires significant effort and time for each one of the steps presented in Figure 3-1. Creation of manual semantic annotations has the potential of high quality results, without any guarantee though. Section 3.3 proves the difficulty and complexity of discovering vocabulary terms and afterwards developing the corresponding annotations based on the result vocabulary. Facilitation of the annotation process is crucial for the realisation of the fundamental Semantic Web vision of a machine-understandable Web of Data. The success of reaching the level of sharing machine-understandable content has outstanding impact in the ways that the content can be used and the services that can be built on top of this metadata rich online Web scale database.

In this scope, already since the beginning of the research initiative in the Semantic Web area, a few approaches were proposed that will allow the semi-automatic development of semantic annotations. Those are summarised below and compared in order to realise what has already been accomplished and which areas have space for further research commitments in order to shorten the distance between the Web data current status and the aforementioned vision.

Erdmann et al. describe in [26], a framework built on top of SMES [59], which aims to facilitate the semi-automatic development of annotations for a given web document. The approach is based on the information extraction that SMES performs on the input document. The extracted content is mapped to the predefined ontology, which is related to the domain of the use case. The contribution presented within this manuscript focuses in the transition from the manual annotations to semi-automatic annotations and the lessons learned in the two types of approaches rather than thoroughly describing a system that performs the semantic annotations generation.

AeroDAML. AeroDAML is a knowledge markup tool based on the DAML ontologies. It applies Natural Language Processing (NLP) to extract information and automatically generate DAML annotations for webpages. AeroDAML links nouns and common relationships with classes and properties in DAML ontologies as described in [47]. The DAML annotations produced by AeroDAML consist of mappings between document entities and classes or properties of the DAML ontology. It is able to annotate named entities and also other nouns by finding the corresponding class in DAML (e.g. 10 EUR instanceOf money, Austria instanceOf nation).

CREAM/Ont-O-Mat. Handschuh et al. introduced CREAM in [36], as a framework to generate annotations, but also as an authoring framework focused on metadata creation. It allows the creatino of metadata but also the creation of relational metadata, which is described as interrelated definition of classes in a domain ontology. Also, CREAM employs a crawler to facilitate the discovery of entities in the Web of Data that can be mapped to proper nouns of the text that has been loaded in the editor of CREAM. Ont-O-Mat is the reference implementation of the CREAM framework.

DBpedia Spotlight. Another Linked Data enricher has been developed by the DBpedia foundation [19] and aims to discover and weave references to entities that are described in a web document. However, the online available tool for demonstration[22], enriches a given text with links to entities from the DBpedia knowledge base, but it does not provide any of the formats described previously. The output is rendered within the input text box and transforms the various words that refer to entities to links to the respective webpage on DBpedia.

GoNTogle. GoNTogle has a different scope than the rest of the approaches as its main target is the annotation of normal documents (DOC, PDF). The annotation of the documents is performed by using OWL & RDFs ontologies, while the annotations are stored within an Ontology Server as ontology instances according to the architecture of GoNTogle, as presented in [30]. The framework consists of an editor that is used to load the document and the ontology that is used for the annotations. Apart from the manual annotation, the approach proposes an automatic mechanism by leveraging the user's annotation history.

KIM. This platfrom focuses on the named entity annotations rather than on the annotation of the webpage content in general as described in [63]. It leverages the GATE framework [18] by employing GATE's document management functionality and NLP components. The task of enriching the target webpage or document with named entity annotations is based on a very large knowledge base, the KIM KB. The knowledge base has been pre-populated with entities from various domains and of general importance in order to facilitate the information extraction process in multi-domain web content. KIM KB consists of more than 80,000 entities, which include a lot of geographical entities, like locations, countries and cities. Apart from the knowledge base, KIM includes an upper-ontology the KIM Ontology that defines

[22] DBpedia spotlight: `https://dbpedia-spotlight.github.io/demo/`

all the entity classes, attributes, and relations. Therefore, the KIM approach can be considered as a semi-automatic annotation solution for the entities of the target webpage. The maintenance of the entities in the knowledge base is the reason that the approach is classified as semi-automatic in the scope of the presented related work research.

KODA - The knowledge driven annotator. Similarly to DBpedia Spotlight, KODA[23] aims to facilitate the annotations of text nodes in a knowledge-driven way by leveraging specific knowledge-base sources. DBpedia is one of the sources that can be used to distinguish named entities and link them to the input text. KODA is based on an unsupervised approach and as stated in the related publication of the authors [57]: *"a fully unsupervised approach that relies only on the KB content and uses no textual patterns, no learning corpora and no prior disambiguation information"*. The result output is HTML with links on the respective text elements.

MnM. Mnm is an annotation tool that integrates a web browser with an ontology editor and aims to facilitate the generation of semantic annotation in both semi-automatic and automatic ways, as it was described in [92]. The tool includes a step of selecting the subset of knowledge components from a library of knowledge models in order to later use them for information extraction of the target documents and annotation of those. Specifically the selected ontology models are used to manually enrich a corpus of documents which will be later used as the training set that will allow to formulate the extraction rules that will be used to annotate the pool of target documents. Therefore, the classification of the approach to a semi-automatic is more appropriate as there are already two steps involved that require manual effort by the users of the tool.

MUSE. The MUSE system introduced in [54], is described as an information extraction system that facilitates named entitiy recognition tasks. It has been based on GATE [18] and includes a significant number of steps, including the extraction

[23] KODA demo: `http://smartdocs.list.lu/koda/demo`

of tokens, splitting of sentences and recognition of the parts of speech (POS). The core functionality regarding the semantic annotations is taken care by the Semantic Tagger module, which consists of grammar rules based on the JAPE language and allows to map grammar patterns to specific types (ontology class terms). The process of selecting the correct classes depending on the text type is automatic and does not require manual effort.

OnTeA. Laclavik et al. introduced in [48] the OnTeA semantic annotation approach, which aims to build semantic annotations in a predefined domain for a given input webpage. As the authors describe the basic idea that resides in the core of the approach, the tool analyses the input document using regular expression patterns to detect semantically equivalent elements according to the predefined domain ontology. In a later iteration of OnTeA, the authors presented in [49] the porting of the approach to the Hadoop ecosystem while seeking high scalability of the approach in large document sets.

PANKOW. Another pattern based approach is presented in [16]. The PANKOW approach has been integrated with the CREAM annotation framework and the OntoMat Annotizer[24]. The approach starts with the extraction of all the nouns in the given document and then continues with the generation of hypothetical sentences by combining the extracted tokens with the domain ontology that refer to the input document. The set of patterns that is being used in this approach is basically syntax based by exploiting *isa-relationships*, references, phrases that define entities by using the article *"the"*, etc. One weak point, yet interesting, of the approach is the integration of the Google search API, which is crucial in order to narrow down the hypothetical sentences. The search API is used to search all the sentences and decide on their correctness based on the number of the search results. Finally, it is highlighted that the interactive annotation of the input documents provide better results, which makes the approach eligible to be classified as semi-automatic.

[24] OntoMat Annotizer: `http://km.aifb.kit.edu/projects/annotation/Members/cobu/AnnotationTool.2004-07-28.1138/view.html`

Within the next iteration of the approach, namely the C-PANKOW as presented in [17], the authors aimed to address any shortcomings of the previous version. The major updates were related to the usage efficiency of framework a) by introducing a new step of downloading the abstracts that are analysed; b) by reducing the requests to the Google search API; and c) by considering the similarity of the downloaded abstracts and the target webpage.

SemTag. The core contributors of SemTag realise in [22] that the existence of semantic annotations resides in a circular dependency with the existence of applications that make use of machine understandable data. Similarly to the goal that this PhD work aims to achieve, the authors try to break this cycle and provide an early set of widespread semantic tags via automated semantic tagging of large corpora. SemTag is based on the TAP ontology by using a Taxonomy Based Disambiguation (TBD) algorithm as described in detail in [21]. The workflow of the approach includes the detection of particular entities based on the TAP ontology that derives from the TAP knowledge base [65]. The TAP ontology contains more than 65,000 instances of common known entities like cities, countries and names of individuals, as described in [24]. In terms of performance, SemTag seems to have a good precision, as the authors have tried to annotate a corpus of 264 million pages, which resulted in 434 million semantic tags with an estimated accuracy of 82%. In addition, the framework is able to handle very large repositories of documents in an autonomous way with only one manual step according to the creators, i.e. the disambiguation of the extracted tokens based on a manually generated corpus of metadata with the verdict about the reference of specific labels to the corresponding entities in a given context. Furthermore, it is important to highlight that the annotations refer to entity tagging within the documents rather than to semantic description of the content. For example, SemTag is able to recognise a city name in a document and tag it accordingly in order to add semantics to the content. However, it does not aim to annotate a document that describes a recipe for example, which needs a large set of properties to be located and annotated.

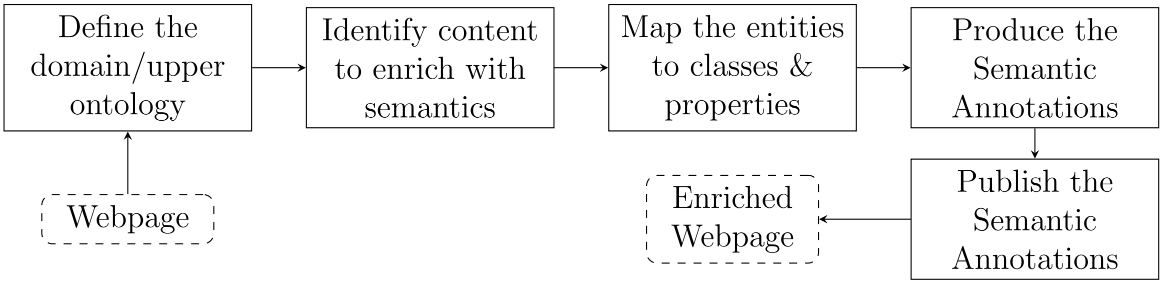

Define the domain/upper ontology
Identify content to enrich with semantics
Map the entities to classes & properties
Produce the Semantic Annotations
Webpage
Enriched Webpage
Publish the Semantic Annotations

Figure 3-8: The semantic annotations generation workflow with the usage of a domain ontology and neglecting the rest of vocabularies in the Web.

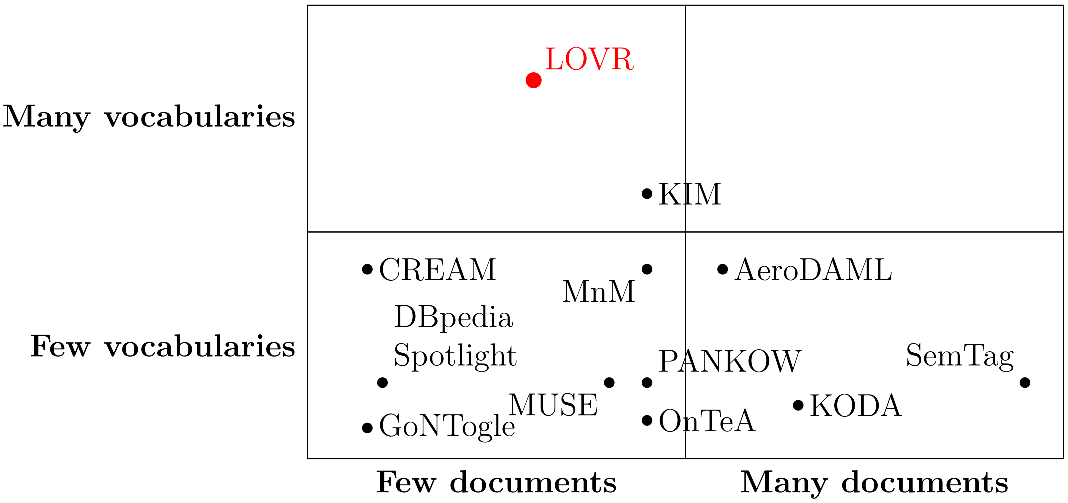


Figure 3-9: Related work projected in a two dimensional diagram with axes *y: number of vocabularies used* and *x: number of documents that the approach aims to annotate.* The LOVR item that appears in the diagram refers to the proposed approach of this thesis.

One of the common characteristics of all the aforementioned related work is the employment of an upper ontology to facilitate the semantic tagging of the target document. Figure 3-8 depicts in a generic way the steps that are involved when a domain ontology resides at the core of the approach. Most of the approaches described above realise the topic of semantic annotations with the named entity annotations as the primary target. That said, their focus is on recognising instances of ontology classes like a *person*, a *city* or *country* and tag the name accordingly to reflect the *is-a relationship*. In addition, leveraging a domain ontology for information extraction (IE) purposes is extensively discussed within the related work.

As shown in Figure 3-9, most of the approaches are classified as methodologies for scenarios that the vocabularies used are specific and predefined. The proposed approach (LOVR/Vocab-recommender) is very different compared to the rest, as it takes in consideration the whole set of available vocabularies in the vocabulary space (LOV) and not only one specific or a limited subset. However, it can be mostly used within the scope of a website and its webpages rather in the scale of the Web. SemTag seems to be appropriate for very large datasets, but it is using a specific upper ontology. The ideal solution would be one that combines the potential of annotating a large corpus of documents by leveraging all the vocabularies that exist in the vocabulary space. All of the approaches were designed before the creation of the Linked Open Vocabularies registry (LOV), which launched in 2012[25]. This fact has significant impact on the design of the approach described within the presented PhD work, i.e. *LOVR* and the reference implementation of it the *vocab-recommender*.

A similar survey to the one conducted in the scope of the PhD and presented above about the facilitation of semantic annotations is presented in [64]. The authors separate the approaches in two main categories, i.e. a) those that are based on patterns and b) those that are based on machine learning. The former refer to sets of rules that are used to discover entities and annotate them, while the latter refer to probabilistic analysis and examination of the linguistic structures. The highlight in the comparison of the two types of approaches is the need or not of a significant amount of already annotated webpages in order to built annotations for the input document. This step is called training of the system and it is a prerequisite for the machine learning based approaches.

In addition to the above perspectives, the discussed approaches, as depicted in Figure 3-9, have been compared from two more perspectives, i.e. a) the amount of vocabularies that the approach can leverage to be exploited in the generation of semantic annotations for a given document and b) the scalability of the approach for large sets of documents.

Studying the above described approaches, it is prominent that the semantic annotations development topic has a lot of years of research record with important

[25] LOV info page: http://lov.okfn.org/dataset/lov/about

achievements and results. However, the topic is still not fully covered and new factors enter the context of the research area offering new opportunities for further developments and innovation. The following list refers to open topics in the field of semantic annotations:

1. Exploitation of the abundance of the vocabularies available in the Web of Data.

2. Exploitation of the schema.org vocabulary.

The focus is on the vocabularies, as significant steps have been made to make them more accessible and reusable by the Web engineering community. On one hand there is schema.org, which is the result of the systematic work by a group of contributors and aims to built a universal vocabulary that covers the majority of domains that appear on the Web and would benefit by a semantically rich and structured representation. Schema.org was presented earlier in Section 3.1.2. On the other hand, we have the rest of the vocabulary space, which contains many vocabularies with significant importance and contribution to the Semantic Web. Before the Linked Open Vocabularies (LOV) initiative, which was presented in Section 3.1.1, it was nearly impossible to explore the vocabulary space, as it was fragmented and hard to discover new vocabularies. LOV changed the vocabulary exploration by providing one single reference point for searching vocabularies and vocabulary terms. Throughout the presented related work, we can distinguish a trade off that exists in the development of semantic annotations. If the approach aims to scale for large datasets then it will most likely leverage one or a limited number of vocabularies as it is easier to match the document content to a limited set of terms rather than to a large collection of vocabularies and terms.

3.5 Vocabulary development collaboration

The development of vocabularies is a task that requires in most of the cases more than one contributors. Including more than one authors in the creation of a new vocabulary ensures the quality of the result by combining the expertise of the stakeholders in different fields. Listing 3.1 presents a SPARQL query that facilitates the retrieval of vocabularies that have been created by more than one contributors. The LOV directory includes two different fields in the vocabulary profile, namely the *creators* and the *contributors*. At the moment of the manuscript writing, the above-mentioned query returned 254 vocabularies out of the 567 of vocabularies in total, which means that 45% of vocabularies in the LOV repository are created by more than one contributors; proving the assumption that a vocabulary in many cases is the result of teamwork and collaboration. The number of total vocabularies mentioned in the previous calculations refers to vocabularies that are enlisted with at least one creator. Apart from those, there are vocabulary entries which do not mention any creator or contributor, i.e. 23 vocabularies. Those are not taken in consideration in the calculations as their metadata is considered incomplete in this scope and the measurement will be biased by those.

```
PREFIX voaf: <http://purl.org/vocommons/voaf#>
PREFIX dcterms: <http://purl.org/dc/terms/>
SELECT DISTINCT ?vocab ?authors {
{
 SELECT ?vocab (count(distinct ?creator) as ?authors) {
 GRAPH <http://lov.okfn.org/dataset/lov>{
        ?vocab a voaf:Vocabulary.
        ?vocab dcterms:creator ?creator. }
 } GROUP BY ?vocab
        HAVING (?authors >1)
        ORDER BY ?authors
}
UNION
{
 SELECT ?vocab (count(distinct ?contributor) as ?authors){
 GRAPH <http://lov.okfn.org/dataset/lov> {
        ?vocab a voaf:Vocabulary.
        ?vocab dcterms:contributor ?contributor.
 }} GROUP BY ?vocab
        HAVING (?authors >1)
        ORDER BY ?authors
}}
```

Listing 3.1: LOV SPARQL query to retrieve vocabularies that have more than one creators or more than one contributors.

In this direction, related research has been conducted by various research groups in order to facilitate the collaboration in the development of a vocabulary. Collaboration has many perspectives in the vocabulary development, including modelling discussions, domain experts input, input by authors of existing related vocabularies, or cooperation of vocabulary engineers from various partners when it is related to a common project task. The survey presented in [75] compares the various methodologies and approaches by evaluating them against a set of criteria, including the types of roles in the approach, the usage of the methodology, the collaboration means, etc. The DILIGENT [20] methodology is presented to be the one that better address all the aforementioned criteria of the survey. Many tools are presented and put under question in the survey, including the popular Protégé[26] and wiki based approaches. Apart from the collaborative communication within the scope of the tool via annotations or communication threads, the ability to handle different versions via a version control system is considered important for a smooth collaboration flow. Neon Toolkit[27] and Protégé provide special plug-ins for the collaboration dimension, respectively the Cicero[28] and the Collaborative Protege[29].

Definition 9 (Taxonomy) ***Taxonomy*** *is an hierarchical representation of entities*

[26] Protégé webpage: `http://protege.stanford.edu/`

[27] Neon Toolkit webpage: `http://neon-toolkit.org/`

[28] Cicero plug-in: `http://neon-toolkit.org/wiki/Cicero.html`

[29] Collaborative Protege plug-in: `http://protegewiki.stanford.edu/wiki/Collaborative_Protege`

that result in a classification based on parent-child relationships.

Futhermore, a lot ot approaches aim to facilitate the editing and maintenance of taxonomies (taxonomy is defined in Definition 9) in a collaborative manner. Taxonomies are very simpler than vocabularies in structure, as they do not include any other relationship types than subclass - superclass. The main feature that they have in common is the relation between terms and concepts, which in both scenarios are used to enrich the content of a document with semantics. However, a taxonomy can grow a lot and the management of it can become cumbersome. In this scope, a few approaches are enlisted here, as part of the collaborative vocabulary development. *SOBOLEO* [93] is a collaboration tool for the development of SKOS based thesauri and with a special focus on the social networks. *PoolParty* [72] is a thesaurus management tool, facilitating the maintenance of taxonomies that are enriched via Linked Open Data and many other sources via the user interface that it provides. *VocBench*, described in [87], facilitates the maintenance of SKOS-based thesauri, focusing on the colloboration aspect of it, which is crucial for the publication of large taxonomies.

More recent approaches have arisen in the meantime that benefit from the newest versioning control system methodologies and with major goal the facilitation of the collaborative vocabulary development. The latest versioning control system methodology, which has changed the field radically, is *Git*[30], that has evolved from the need of managing very large projects like the Linux Kernel. An approach that showcases how Git can be adopted to vocabulary development is described in [34], namely the *Git4Voc* methodology. On top of that, the *VoCol* approach presented in [35], has been later designed taking in consideration the experience gathered during the research endeavours for *Git4Voc*, aims to a more holistic approach by introducing a quality assurance dimension which is responsible for the generation of a report about the compliance, in terms of domain requirements fulfilment, of any change to the tracked vocabulary via a monitoring service for repository changes. In addition a syntax validation ensures that the committed change results in a valid vocabulary.

[30] Wikipedia for Git: `https://en.wikipedia.org/wiki/Git`

3.6 Summary

This chapter aims to provide a presentation of the related topics to the proposed approach. Therefore, it started with an introduction to the process of generating semantic annotations and the tools that are available to support the engineer at the time of this thesis. In this scope, a presentation of the *LOV* repository is provided as part of the vocabulary exploration together with other approaches, like the *vocab.cc* and *LODStats*. Special part of the section has been dedicated to schema.org, which because of its nature is considered to have significant importance in the Semantic Web realisation.

In addition to the exploration of vocabularies via the various repositories of Linked Data, or vocabularies, the vocabulary ranking is discussed from the prism of assisting the discovery of vocabularies. Various approaches are presented with *DWRank* and the *LOV* ranking to be the main sources of inspiration for the ranking that has been designed in the scope of this thesis.

The complexity of defining the semantics in an informational piece of text has been proved by the survey provided in Section 3.3, which aims to examine the amount of time needed by an evaluator with minor experience in semantic annotations generation, as well as to capture any patterns that occur throughout the process. Domains that are already structured within the webpage to some extent, like the recipe presentation, seemed to be an easier task across the pool of evaluators of the survey. A Web document about a recipe has some specific parts, like a list of ingredients and execution steps that can easily be grouped together and interpreted as parts of a recipe entity. On the other hand, a Web document of type article belongs to a more fuzzy domain to annotate, as we could consider the various provided information as parts of the entities that are important for the webpage and should be declared in the semantic mappings apart from the article itself.

At the core of the related work together with the vocabulary ranking approaches and the manual discovery of vocabulary terms, an extensive selection of semi-automatic semantic annotations generation approaches is presented and compared. The goal is to define what directions have been addressed and which are the open issues in the field. Most of the presented approaches are based on a very limited set of vocabulary terms most of the cases within the same vocabulary. Furthermore, the directions that have been recognised as open for further research include: a) exploration of the vocabulary space to annotate Web content and b) the exploitation of the schema.org vocabulary.

Finalising exploration of the related work map, the collaboration aspect is discussed within Section 3.5, which is also considered important for the pragmatic realisation of the semantic annotations development. Defining a vocabulary is not an easy task and, as presented in the abovementioned section, most of the vocabularies in the vocabulary space is the result of the collaboration of many authors.

For each one of the various steps presented in Figure 3-1, there are obstacles and respective approaches that aim to facilitate the process realisation as discussed within this section. The proposed approach contributes mainly to the second step, i.e. the discovery of candidate vocabulary terms, as it will be discussed later in Chapter 4. Combining the information that can be distilled from the various aforementioned services and resources, the proposed approach discussed through the Chapters 4, 5 and 6 achieves to play the role of the term discovery assistant.

Chapter 4

Theoretical basis of the approach

Towards the transformation of websites to APIs

Moving beyond the state of the art, the aim of the proposed approach is to leverage a website to a self-described Application Programming Interface (API), ready to be used as a structured data source without any further integration effort by the prospective stakeholders. An API in principle is a contract with the users of it (humans or machines) on how to interact with the corresponding entity via the exposed interface. Therefore, it should provide sufficient information about any type of data exchange that is needed in order to enable the communication with the underlying entity within the functionality borders that the interface owner decides. For example, an API would provide functionality for information retrieval (reading); information submission (writing) or requests for deleting resources at the main entity side.

In this scope, the above fundamental operations need to be mapped to methodologies that will substitute the functionality that a regular API would provide via the various endpoints. Thus, this chapter examines which methodology could enable the lifting of a website to a self-described API for the presented information. The first step is to make the content understandable by machines in an autonomous manner; without human intervention. Making the presented content machine-interpretable requires the explicit definition of the semantics of the presentation information bits. In other words, the meaning of the content should be explicitly declared following an approach that the search engines would be able to identify, but also any stakeholder that would like to interact with the business entity via a programmatic interface. The thesis of the presented approach is that ***the generation of semantic annotations for a given webpage suffices to transform the content to meaningful bits for agents (crawlers, data consumers)***. At this point, it is important to clarify that the approach is not limited to make a website more understandable for search engines (e.g. Bing, Google, Yahoo!, or Yandex), but to further transform it to a comprehensively annotated web entity that could seamlessly integrate in the Web of Data.

The chapter starts with the comparison of a website to an API in Section 4.1 in order to examine the overlapping of data between them, which later discussed within Section 4.2 in order to outline the proposed approach. Section 4.3 presents the recommendation base in terms of sources that can be employed in the proposed recommendation methodology. Furthermore, in Section 4.4, the search of vocabularies is analysed including the ranking of vocabularies. Similarly, Section 4.5 focuses on the search and ranking of vocabulary terms. The rest of the chapter, provides a discussion on the various presented dimensions in Section 4.6 to summarise the chapter.

4.1 Comparing websites to APIs

Reflecting the title of the section, the main aim of this part is to put in comparison the nature of websites and APIs following the introduction of the chapter. In this scope, a classification of websites has been designed for the needs of the presented thesis. The taxonomy is based on a single criterion, i.e. the way that the information is presented. We could follow a dichotomy by dividing the websites into two big buckets, those that are mainly static and those that are dynamic in a similar way that we have classified the dissemination in [27]. As it is presented in that journal article, the dynamic dissemination can be split further into sub-categories. Similarly, in order to bring the classification closer to the needs of this thesis, the following categories are defined: a) simple presentation websites, which includes company profiles and

Category	Example websites
a. Simple presentation	hawking.org.uk
b. Blogs, news websites	mashable.com, bbc.com/news
c. Online shops	amazon.com, zappos.com, zalando.de
d. Search engines, directories	google.com, bing.com, yahoo.com, yandex.com, kayak.com, imdb.com, yelp.com
e. Social networks	facebook.com, twitter.com, linkedin.com
f. Sharing platforms	instagram.com, flickr.com, youtube.com

Table 4.1: Websites classification example.

personal websites mostly, b) blogs and news websites, which includes in general information structured around the concept of the article, c) online shops, d) search engines and directories, which includes broad search engines like Google, bing, Yahoo!, but also search engines for specific verticals (e.g. a doctors directory); e) social networks and f) sharing platforms for the various multimedia types. This classification does not aim to be exhaustive and comprehensive by finding a bucket to assign any possible Web application, but aims to support and showcase the various categories on which the presented approach can have an impact. Table 4.1 shows the six categories accompanied with a few examples of websites that they include.

In principle, an API could implement functionality that falls back to one or more methods from the basic set of create, read, update, delete (CRUD) operations. In a similar way, the Representational State Transfer (REST) paradigm exploits the HTTP methods in order to implement the four aforementioned operations for the needs of the communication of a Web server with the outer world. the term REST was coined by Roy Fielding[1] in [29]. The most interesting design aspect of the architectural paradigm is the exploitation of the basic HTTP methods to reflect the various CRUD operations. In a nutshell, to create a resource on the server the endpoint should be configured to be used with HTTP POST, to retrieve a resource would be done via HTTP GET, to update the state of a resource should be used the HTTP PUT and to delete a resource it should be performed via an HTTP DELETE.

From the above described processes, the read one is considered the most basic

[1] REST definition: `http://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm`

and fundamental for the interpretation of a website as a data endpoint. As declared by Axiom 1, a website can be considered as machine understandable if and only if it can be consumed autonomously based on a contract. The contract in the proposed approach between the provider and the consumer is the vocabulary that will be used and it is a common publicly published set of terms with specific semantics.

Axiom 1 *If a website is formatted in a way that can be consumed autonomously based on a contract, then the website supports the read operation and can be considered as machine understandable.*

All the website categories of Table 4.1 have one thing in common, they provide a representation of the underlying data independently of the fact if it comes from a database, a documents storage or it is hardcoded as a static HTML page. Simple presentation websites, which is the first category in Table 4.1, refer to websites that are either static or even if they are based on a Content Management System the presented content is not changing often. On the contrary, all the rest of the categories refer to websites that the presented data is constantly changing and for some of them the speed is really high, e.g. on social networks the users produce content in great volumes per minute. However, no matter the category, the presented data could be extracted using a crawler that understands a finite set of vocabularies and the semantics of the extracted values would be explicit. Given this ability to consume the website data, it would not add substantial value to implement a separate API that would be queried in order to retrieve information.

Would it be possible in many cases to avoid the implementation of an API in order to make the website content accessible to third parties? An example could be the metasearch engine model which needs to integrate hundreds of different APIs in order to provide a summarised comparison about the purchase alternatives of a buyer. For example, idealo[2] in Germany or skroutz[3] in Greece need to come in agreement with the various e-shop owners in order to build a communication path to receive the product entries from the various respective retail e-shops. Then, they

[2] idealo: `http://www.idealo.de/`

[3] Skroutz: `http://skroutz.gr`

API Operation	Website equivalent functionality
Read	Semantic annotation
Create	schema.org AddAction[7]
Update	schema.org UpdateAction[8]
Delete	schema.org DeleteAction[9]

Table 4.2: Mapping of API basic operations to website equivalent functionality.

are able to receive product information through this channel and present them to the search results of the users for consumer products. However, in case the e-shops were able to provide a product inventory on their website with explicit semantics of the presented data following a public vocabulary, then the metasearch engines would be able to parse and extract the needed information in a unified way without the need to integrate a separate API for each e-shop. In a similar way, any business owner could integrate with partners via a much easier process and less costly. This would open up new opportunities for all the stakeholders. Business to business (B2B) relationships would be easier to establish by facilitating the exchange of data between the two sides. Similarly, the metasearch travel engines KAYAK[4], skyscanner[5], trivago[6] need to integrate specific APIs for the various providers of travel products in order to be able to build a search engine on top of the pool of itineraries and options. However, if the structured data paradigm based on semantics was embraced by the travel providers, then it would be easier for the metasearch engines to integrate those and also from the providers side it would be easier for them to be searched by the metasearch engines and appear on the search result pages.

The approach introduced in Section 4.2 aims to bridge the aforementioned gap by providing a methodology that can be used to leverage any website to a ready for external interactions website. The methodology mostly focuses on enabling a website to provide the read operation from the CRUD set discussed at the beginning of the section. However, later in Section 5.4.2, one more dimension of the methodology is discussed, which is related to the combination of the Web entities with possible

[4] KAYAK: `https://www.kayak.com/`

[5] skyscanner: `https://www.skyscanner.com/`

[6] trivago: `http://www.trivago.com/`

actions. The schema.org vocabulary has started to provide a set of vocabulary terms that can be used in order to describe actions that the presented entity on the webpage can perform. In this respect, Table 4.2 describes the API basic types of operations and how they map to functionality that can be added on webpages when the data is formatted using semantics that describe concrete entities.

4.2 Proposed approach description

Three main parts comprise an approach description: a) the purpose of the approach; b) the process needed to make use of the approach; and c) the workflow behind it. These three aspects are covered through out this section in order to provide to the reader a description of the objectives that the proposed approach aims to accomplish, a general understanding of the underlying workflow and the main usage scenarios.

4.2.1 What is the purpose of the approach?

Through out the previous sections, the manuscript discussed the various dimensions related to the Web of Data and the Semantic Web technology accomplishments regarding the layer of metadata added on top of the various data sources. Furthermore, the previous section (4.1) briefly analysed the main differences of a website and an API, which proves the assumption that a website could play the role of an API up to an extent by substituting the fundamental operations. There is a part of functionality and data that is shared between the exposed content via a website and via an API, although the existing implementation paradigms tend to separate by assigning a different usage profile to each one of them.

The proposed approach aims to bring those two sets of data presentation functionalities closer by providing a process that could be followed in order to transform the website content to programmatically consumable entities. In this respect, the website will be machine understandable and will have been leveraged to an API, without developing any new endpoints. Let $\mathcal{W}$ be the set of content presented by a website and $\mathcal{A}$ be the set of content exposed through the API of the same website. In this

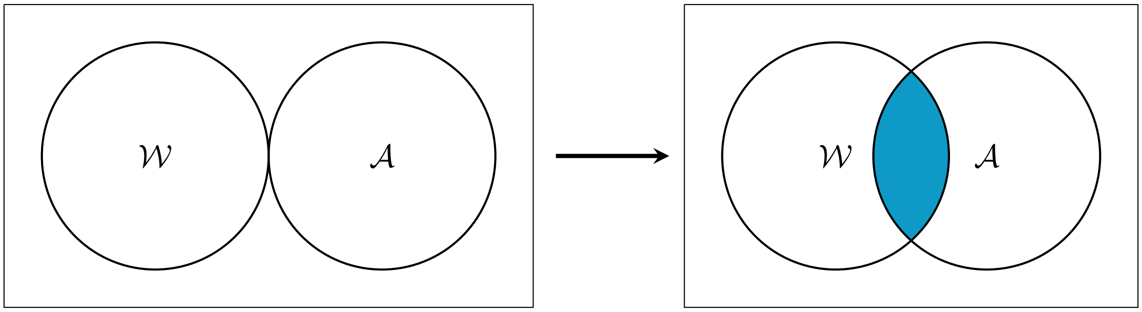


Figure 4-1: Venn diagram of website and API functionality. The proposed approach aims to bridge the gap between websites and APIs through semantics. The left venn diagram shows that websites are not used as API endpoints, while the right venn diagram demonstrates through the overlapping between the two concepts that the annotated websites can function as APIs at a significant level.

respect, Figure 4-1 visualises how the proposed approach alters the venn diagram of a website and the corresponding API. It achieves the transformation of a significant part of the already presented data to a source of structured data similar to what an API endpoint would provide.

4.2.2 What are the benefits of the approach?

The above mentioned idea is the ultimate aim of the proposed approach and the main motivation behind it. In addition, the approach *facilitates the discovery of vocabulary terms by playing the role of an assistant*. For a given webpage, let $\mathcal{W}$ be the set of keywords with size $|\mathcal{W}| = n$, $n \in \mathbb{N}$, then for each keyword w_n there is a set of results R_{w_n} with size $|R_{w_n}| = m$, $m \in \mathbb{N}$. Thus, for each keyword the vocabulary engineer of the webpage developer needs to perform a search and decide on the top terms from the total of m results when using a directory like the LOV search. This process needs to be reproduced for n times, as there are n keywords to be checked. From the total $n \times m$ matrix of keywords, the user needs to narrow down to n terms which will be used to describe the content of the webpage. Therefore, the complexity and time cost for this process is considered significant. This challenge is addressed by the proposed approach by automating the process and providing a set of vocabulary terms to consider for the annotation of the content.

Furthermore, the presented approach aims to *educate engineers and Web developers by introducing them to the possible terms that can be combined with the given webpage content in order to provide a semantic representation of it.* Facilitating the discovery process allows to smooth the learning curve and introduce the engineers to the semantic annotations topic. The outcome is positive both for the engineers themselves as they are assisted in their objective, but also for the uptake of the semantic annotations paradigm, as it reduces the users that decide to not move forward with the semantic annotations due to any possible difficulties they face in the discovery process.

Additionally, users can theoretically *ask targeted questions on the data of the webpage* by using various browser extensions, like RDF Triple Collector[10], RDFa Developer for Firefox[11] or any other implementation following the idea of the RDFa API[12].

Business-to-business relationships can benefit from the inclusion of semantically annotated sructured data in the websites, as it *eliminates the need of building an additional API.* A website should be enough to describe the data of an organisation, which allows the organisation to save resources. The case of a small and medium-sized enterprise (SME) would benefit in the integration of its services with a partner by avoiding the costs of building special interoperability support solutions for each one of the partners.

Finally, *facilitating the annotation of websites with specialised vocabularies beyond schema.org, enables search engines that focus on a specific domain, like food, travel, etc. to better make sense of the data of the relevant websites.* In contrast to the general purpose search engines, like Google and Bing, the specific search engines would be interested in more detailed description (annotation) of the presented data with vocabularies that are targeting the domain under question. In this case, the approach facilitates the selection of vocabulary terms that can be leveraged to describe

[10] RDF Triple Collector: `http://www-sop.inria.fr/members/Fuqi.Song/rtc/rtc.html`

[11] RDFa Developer Firefox add-on: `https://addons.mozilla.org/en-US/firefox/addon/rdfa-developer/`

[12] W3C RDFa API for extracting structured data: `https://www.w3.org/TR/rdfa-api/`

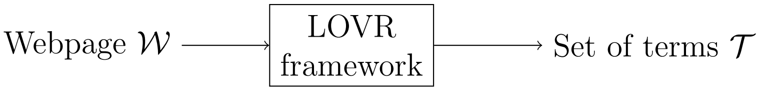


Figure 4-2: Usage scenario of the LOVR framework.

the structured data of a website and produce semantic annotations.

4.2.3 How is the approach used?

The overall workflow of the proposed approach, depicted in Figure 4-2, starts by giving the target webpage at the input and receiving at the output a set of vocabulary terms from the LOV space. The set $\mathcal{T}$ of terms could be used by the website developer to enrich the content with vocabulary terms that will transform the website data to machine understandable entities.

The result set is not a simple list of terms, but a justified list with the accompanied keyword that yielded the results and the respective ranking score. In this way the result set transparently reflects the reason that it was constructed as such. Comparing the input with the proposed terms, the user of the methodology is able to educate herself about the various vocabularies that exist and their usage potential. Therefore, the user is not aware of the various steps that need to be executed against the LOV repository, which helps to introduce her to the semantic annotations topic without feeling any confusion about the process of discovery. However, as soon as a user of the approach has informed herself about the process, she could herself use the LOV search to discover terms and manually follow the steps of the methodology in order to follow different directions to accomplish the given task.

4.2.4 What is the workflow of the approach?

The workflow of the approach is composed by all the necessary steps to map the webpage content to a set of vocabulary terms. Figure 4-3 demonstrates the various steps in a flow diagram. The very first step includes the extraction of the keywords that reflect the content parts that need to be annotated. Those keyword tokens are the input to the search within the LOV directory about related terms from the listed

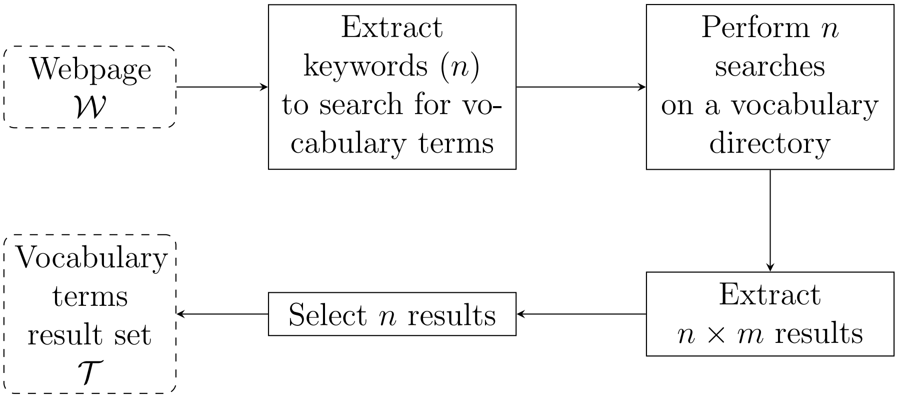

Webpage
$\mathcal{W}$
Extract keywords (n) to search for vocabulary terms
Perform n searches on a vocabulary directory
Extract $n \times m$ results
Select n results
Vocabulary terms result set
$\mathcal{T}$

Figure 4-3: Using a vocabulary directory for vocabulary terms discovery.

vocabularies. This search enables both the discovery of vocabulary terms, but also the exploration of LOV metadata about the vocabularies. For each keyword the result set is ranked and the top terms are added to the result set of the approach. The various vocabulary term ranking metrics, that have been designed in the scope of the presented PhD, are computed at this stage and employed at the ranking. The ranking scores are those that shape the final set of results that is recommended for usage to the user. In a nutshell, the workflow of the approach is very similar to the manual steps that a user would need to perform, although there are some additional layers regarding the ranking and the presentation.

In this scope, the proposed approach aims to transform the above described workflow to the simplest one depicted in Figure 4-2. Therefore, the various activities of the flow of the diagram in Figure 4-3 are orchestrated and performed solely by the proposed methodology. Combining the manual with the automatic approach a user can develop a better understanding of the vocabulary space and also have an assistant in the process.

4.3 Vocabulary discovery base

The objective of the recommendation layer is to combine the various inputs and provide a set of terms at the output. The recommendation is based on the various metrics that are gathered via the various resources that are integrated at the knowledge base of it. The various sources are very different to each other and provide different metrics and statistical data regarding the usage of vocabularies and the usage of the vocabulary terms. This section introduces the various possible source types that are considered by the proposed approach.

Vocabulary search. As presented by the flow diagram of Figure 4-3, at the core of the discovery is the ability to search for vocabularies and vocabulary terms within a repository that has aggregated all of them at one place. Trying to explore the vocabulary space using general purpose search engines is not efficient as the results will be polluted by irrelevant entries and also the representation and description of the basic information about the vocabulary will be subject to the layout of the corresponding page. On the other hand, using a vocabulary repository with a built-in search endpoint is the ideal source for the proposed approach. As described in section 3.1.1, the Linked Open Vocabularies (LOV) initiative presents a snapshot of the current network of vocabularies, by demonstrating how they refer to each other and which terms comprise them. Therefore, the LOV search is considered the main resource for the vocabulary search functionality.

Vocabulary terms search. In addition to the search of vocabularies, searching for terms is even more important as this type of search is the one that will be leveraged in order to explore the possible answers to the keywords that have been extracted from the input webpage. The LOV search, apart from the vocabularies search, provides a vocabulary terms search, which can be used to extract a list of candidate answers to a given keyword. A few other services are also considered important sources of searching for vocabulary terms, like the *vocab.cc* and the *LODStats*, that were analysed in Section 3.1.3 and Section 3.1.4 respectively. However, these services are considered more informative in the ranking processes rather than the initial discovery part, due to the lack of comprehensiveness when it comes to the documentation of vocabularies.

DBpedia entities. DBpedia due to the structured information representation could play the role of the domain expert in the approach by allowing the retrieval of the most important properties from entities that are in the same domain as the target website. These properties can be used to guide the selection of terms from the various short-listed vocabularies. Thus, in other words it could drive the proposition of terms about the topic of the webpage by leveraging the various properties of the entities in DBpedia as an addition or replacement to the extracted keywords from the webpage itself.

Similar websites. Furthermore, another potential knowledge source for extraction of structured data related to the target website could be the similar websites that already include semantic annotations in their pages. This data is constructed by humans and it could be considered that it reflects the domain expertise of the developer. On the other hand, it is also possible to have mistakes by wrongly interpreting and using the vocabulary terms and all the rest of the issues that we have already discussed in Section 1.1 based on the findings presented in [42]. Furthermore, it is expected to see a limited use of vocabularies as the most used vocabulary would be the schema.org or microformats, which are in the metadata generation field for a while, but irrelevant to the presented approach. Evidence about the trend of the schema.org usage are the quantitative metrics evaluated in the survey that was conducted as part of the presented research work; described in Section 3.3.

Human input. Apart from the various datasources outlined above, the recommendation approach could incorporate feedback from the users that receive and consume the suggestions of the methodology algorithms. The users are encouraged to correct and choose from a closed list of alternatives what they find to be the best fit. This feedback loop enables the methodology to build a collective intelligence layer that improves the recommendation process by employing human input. Input webpages are classified into categories and the interaction with the methodology users can be recorded in a structured way that can be used in future recommendation sessions. This approach has been inspired by the collaborative filtering approach in the recommender systems regarding user's behaviour recording and usage in future sessions by using user's profile similarity [68]. In the scope of the completed research work, we have already presented a part, in Section 3.3, from the respective survey about generation of semantic annotations. The survey aimed to evaluate the difficulty of discovering vocabulary terms for a given webpage and therefore requested from over 50 participants to search for terms for four different webpages. This human input could be the starting point for considering human generated data for the methodology.

Dictionaries. Various sources of dictionaries are taken in consideration to facilitate the extraction of synonyms and addressing of other languages than English. BabelNet[13] and Wikidata[14] are examples of that sources. BabelNet [58] integrates WordNet, Open Multilingual WordNet, Wikipedia, OmegaWiki, Wiktionary and Wikidata, and therefore is a good source for having encyclopaedic and lexicographic coverage of terms in the methodology.

The DBpedia entities would allow the generation of the initial keyword set by extracting the main properties from DBpedia entities similar to the one described in the input webpage. For example, a hotel entity in DBpedia[15] could be useful to extract the amount of properties that are important to be annotated for a hotel in general. Those property names could be used as the input set of keywords to search for vocabulary terms. In the above example of a hotel business entity, the extracted property set would include: hotelname, location, latitude, longitude, number of stars, number of rooms, thumbnail, etc.

The linguistic dictionaries would enable the mapping of other languages to English in order to be able to run the discovery process in a multilingual direction. For example the process of extracting terms from a different than English webpage, would be followed by a translation of the keywords to English to normalise them in order to be able to further search those in the vocabulary repositories, where the terms are described in English.

In addition, similar websites could be employed to extract a set of vocabulary

[13] http://babelnet.org

[14] https://www.wikidata.org

[15] Example of hotel in DBpedia: http://dbpedia.org/page/Austria_Classic_Hotel_Wien

terms that the engineer of the webpage decided to use for the annotation of it. This approach would rely on the assumption that the engineer of the webpage understands both the business domain of the page but also the vocabulary space and has developed semantic annotations that are enough reliable to be used in similar Web entities.

From the aforementioned resources, the proposed approach mainly utilises the vocabulary search and the vocabulary terms search in order to provide recommendations for a set of keywords, which are representative for a given webpage. The set of keywords that has been assembled is not on the focus of the approach and it could be either provider by the DBpedia resources as described before or by a Natural Language Processing (NLP) methodology that would assist the extraction of keywords from the webpage.

4.4 Vocabulary search and ranking

Many different types of search could be employed for the exploration of vocabularies as briefly stated earlier in this chapter. However, the most comprehensive and easy way of searching for vocabularies is the usage of directories like the Linked Open Vocabularies (LOV) service. This particular service searches using full text search within more than 500 vocabularies to find terms that are relevant to a given keyword. The results are returned ranked based on the popularity within the LOV ecosystem, the term popularity in datasets and the label property type the searched term matched [90]. Fundamental part of a Web search is the ranking of the retrieved results. Any type of search in the Web has two main parameters that define the success of the returned results, i.e. precision and recall. Precision refers to the fraction of retrieved instances that are relevant, while recall is the fraction of relevant instances that are retrieved[16].

Definition 10 (Vocabulary search precision) *If $V_{relevant}(v)$ represents the set of vocabularies that are relevant for a keyword v, $V_{retrieved}(v)$ is the number of vocabularies retrieved for Sv, then precision of the vocabulary search is the fraction of retrieved*

[16] `https://en.wikipedia.org/wiki/Precision_and_recall`

vocabularies that are relevant to the keyword v:

$$precision = \frac{V_{relevant}(v) \cap V_{retrieved}(v)}{V_{retrieved}(v)}$$

Definition 11 (Vocabulary search recall) *If $V_{relevant}(v)$ represents the set of vocabularies that are relevant for a keyword v, $V_{retrieved}(v)$ is the number of vocabularies retrieved for Sv, then recall of the vocabulary search is the fraction of relevant vocabularies that have been retrieved for the keyword v:*

$$recall = \frac{V_{retrieved}(v) \cap V_{relevant}(v)}{V_{relevant}(v)}$$

Precision and recall described in Definition 10 and Definition 11, respectively, are also used later in Chapter 8 to measure the proposed results of the methodology in comparison to the results that we would expect to be recommended (or to the results of the manual discovery conducted by the participants of the survey presented in Section 3.3).

Since the first days of the Web, search engines faced the need of applying a ranking methodology for every keyword search against the constructed index. The returning result set needed to be ranked following some criteria that would provide helpful answers to the user for the input keyword. One of the most popular initiatives in this direction is the PageRank algorithm by the founders of the Google search engine, which was firstly introduced by Page et al. in [60]. In this research work, they describe a recursive algorithm that takes in consideration the incoming and outgoing links of a webpage and their ranking at that point of time. One of the characteristics that made the algorithm practically applicable in Web scale is the recursive nature for the computation of the webpages scores and in addition the ability to calculate the score of a webpage without having calculated the score of the interlinked webpages. The algorithm after a few iterations converges to the PageRank of a given webpage as simulation examples prove [67].

Ranking vocabularies is definitely different from ranking webpages for many reasons: a) the amount of vocabularies can be considered finite and not exceeding the 1000 in the near future[17] in contrast to the Web space that is considered infinite in the design of algorithms due to the magnitude of available webpages and the velocity of new webpages creation; b) the ranking aspects are different, while having one common dimension, i.e. the consideration of incoming/outgoing links; and c) vocabularies ranking could affect and be affected by the vocabulary terms ranking, while in the webpages ranking there is not any similar dimension. On the other hand the ranking of webpages is not as simple as considering incoming and outgoing links, but every search engine provider has designed and keeps evolving a sophisticated algorithm that gives scores to the various webpages based on many factors.

The vocabulary search differs from the vocabulary terms search and discovery, while they sound very similar. The latter potentially includes the former, but in many workflows it would be skipped as the focus is on the search of terms rather than of a vocabulary. In the presented approach, the vocabulary ranking reflects the position of the vocabulary within the space of vocabularies and not in conjunction with a keyword search.

The definitions of this chapter aim to describe a few basic terms together with the proposed metrics that are used throughout the presented methodology. An *inactive vocabulary*, as defined in Definition 12, is penalised in the approach as it is not considered a good candidate due to the reason that it has not been used in any datasets or the authors seem to have abandoned it. There are high chances that not maintained vocabularies will not be updated in the future, and any needed improvements will not be scheduled.

Definition 12 (Inactive vocabulary) ***Inactive vocabulary*** *is considered a vocabulary that has not been used by other vocabularies or datasets. Reasons to classify a vocabulary obsolete are: a) a broken hosting page; or b) having a creation date older than a year and without any usage in the LOD datasets.*

[17] The LOV directory has 568 vocabularies as of the September 4th, 2016.

Definition 13 (Result vocabulary) ***Result vocabulary*** *is considered a vocabulary that has been generated by combining vocabulary terms from various vocabularies in order to cover the needs of a specific Web entity.*

Definition 14 (Web entity) ***Web entity*** *is any object on the Web that describes some specific information. A webpage, a website including many pages or an object that is part of a webpage are all Web entities.*

The metric described in Definition 15 measures the relative importance of a vocabulary within a vocabulary space graph by relying on the references to it. The metric is defined in an agnostic way to the vocabulary directory implementation that can be applied on.

Definition 15 (LOV rank) *If B_v is the number of the backlinks to the vocabulary v of the vocabulary space $\mathcal{V}$, i.e. $v \in \mathcal{V}$, then the B_v is divided by the total number of vocabularies to represent the ranking of the vocabulary in $\mathcal{V}$:*

$$VR(v) = \frac{B_v}{|\mathcal{V}|}$$

Applying the above mentioned formula to the LOV repository in order to provide an example, the B_v would be represented by the "incoming links" of the vocabulary LOV profile page[18]. According to Definition 15, B_v is divided with the number of available vocabularies, which makes the range of the metric to be $VR(v) \in [0, 1]$. Therefore, the number of the backlinks of a vocabulary are proportional to the vocabulary ranking score, which assumes that more central vocabularies in the $\mathcal{V}$ graph are better accepted by the community as being more effective than the rest.

In Table 4.3 a few popular vocabularies are evaluated against the aforementioned metric by using the LOV repository as the vocabulary space $\mathcal{V}$ instance of it. At the time of the manuscript authoring, the number of the registered vocabularies in

[18] The FOAF LOV profile page: `http://lov.okfn.org/dataset/lov/vocabs/foaf`

Vocabulary v	B_v	$VR(v)$
dbpedia-owl:	7	0,01
dcterms:	403	0,78
event:	36	0,07
foaf:	307	0,60
gr:	37	0,07
og:	0	0,00
schema:	42	0,08
sioc:	20	0,04
skos:	83	0,16
vcard:	10	0,02

Table 4.3: LOV Vocabulary ranking examples (published in [79]).

LOV is $\mathcal{V} = 512$, which is used to calculate the figures by applying the formula of Definition 15. Vocabularies with terms that cover common concepts, like address and name, are those that score higher in Table 4.3. For example *foaf:* that is widely used to describe personal details, while others that are more specialised do not have that many incoming links as it is difficult to reuse by extension.

In the scope of the proposed approach, an additional dimension in the ranking has been introduced, as presented in [80], which aims to address the "cold start" problem regarding the score of vocabularies that are new in the ecosystem of the Linked Open Vocabularies. As discussed in the same publication ([80]), the idea behind the new metric is based on the assumption that a vocabulary created by authors that have already contributed to a well received vocabulary has higher probability of being accepted by the community and broadly used by the vocabulary engineers. Therefore, a newly introduced vocabulary will be favoured over another one if and only if the authors of it had created in the past vocabularies with a good ranking score. In this respect, the issue with new vocabularies that has been observed in all the existing approaches can be addressed in case the vocabulary contributors have some background that the described methodology can leverage. Since the beginning of the conducted research work, a recognised weakness of the formulas was related to the existence period of the vocabulary, which is leading to implicitly penalising those vocabularies that are newer in the vocabulary space, and probably less used in the

LOD cloud and less reused by other vocabularies.

Thus, introducing the author as the common ground between two or more vocabularies allows the approach to promote newly created vocabularies by authors that have provided vocabularies in the past with a proved quality level. Definition 16 reflects the above described approach and allows to overcome the cold start issue of a newborn vocabulary in the LOV space by giving a score equal or higher than it would be assigned if the author metadata was not considered in the equations.

Definition 16 (Vocabulary author score) *If V_a represents the set of vocabularies that author a has a role in, A_v represents the set of authors of vocabulary v, $VR(v_k))$ refers to the score of vocabulary k based on the incoming links, and $V_{a,i}$ is set of vocabularies related to author i, then let S_v be the score of vocabulary v as defined by:*

$$V_a = V_1, V_2, ...V_m, m \in N$$

$$A_v = a_1, a_2, ...a_n, n \in N$$

$$S_v = \frac{1}{|A_v|} \sum_{i=1}^{n} \frac{\sum_{k=1}^{m} VR(v_k)}{|V_{a,i}|}, n = |A_v|, m = |V_{a,i}|$$

In brief, the formula in Definition 16 calculates the average of the scores of the various vocabularies for each of the authors of vocabulary v and then it produces the score for v as the average of the sum of all the cumulative scores per author. Therefore, the new ranking metric for a vocabulary v has changed and follows the equation of Definition 17.

Definition 17 (Vocabulary rank biased by authority) *If B_v represents the number of vocabularies that are incoming for v, i.e. linking to it, $|\mathcal{V}|$ is the total number of vocabularies, and S_v is the score based on the authors for v, then let $VSR(v)$ be the score of vocabulary v as defined by:*

$$VSR(v) = \frac{B_v}{|\mathcal{V}|} + S_v$$

Vocabulary V	$VR(v)$	$VSR(v)$	$\Delta VR(v)$
dcterms:	0,830	1,140	0,310
foaf:	0,590	0,883	0,293
skos:	0,360	0,578	0,218
schema:	0,090	0,266	0,176
vcard:	0,030	0,058	0,028
event:	0,070	0,096	0,026
gr:	0,070	0,083	0,013
dbpedia-owl:	0,030	0,030	0,000
og:	0,000	0,000	0,000
sioc:	0,040	0,040	0,000

Table 4.4: LOV Vocabulary ranking examples of the old and the new ranking scores in comparison; ranked by the difference in descending order.

To evaluate the effectiveness of the defined formula, randomly selected vocabularies from the LOV dataset are compared based on the score S_v produced by the new formula and the score they would get without this dimension in the calculations. Table 4.4 depicts the comparison results by providing the vocabulary URI at the leftmost column, the default score at the next column and the score taking into consideration the authors at the rightmost column.

As we can see from the data of Table 4.4, there are vocabularies that did not improve in the ranking score with the new aspect. For example the *og* vocabulary is still ranked very low as the authors of it do not appear in any other vocabulary. However, the score of *event*, *dcterms*, *foaf*, *schema*, *vcard* has improved and especially in the case of *event* we consider it to be a significant difference that could help the terms of it to appear higher in the ranking of a result set.

In total, the ranking for 72% of the vocabularies has been affected, i.e. 429 of the 590 vocabularies. Figure 4-4 shows how the frequency of vocabularies being affected are distributed accross the issued year of the last version of the vocabulary. Apparently, from 2007 until 2013 the impact is bigger compared to those closer to the current year, i.e. 2017.

Leveraging the definitions 10 and 11 to the equivalents for the presented rec-

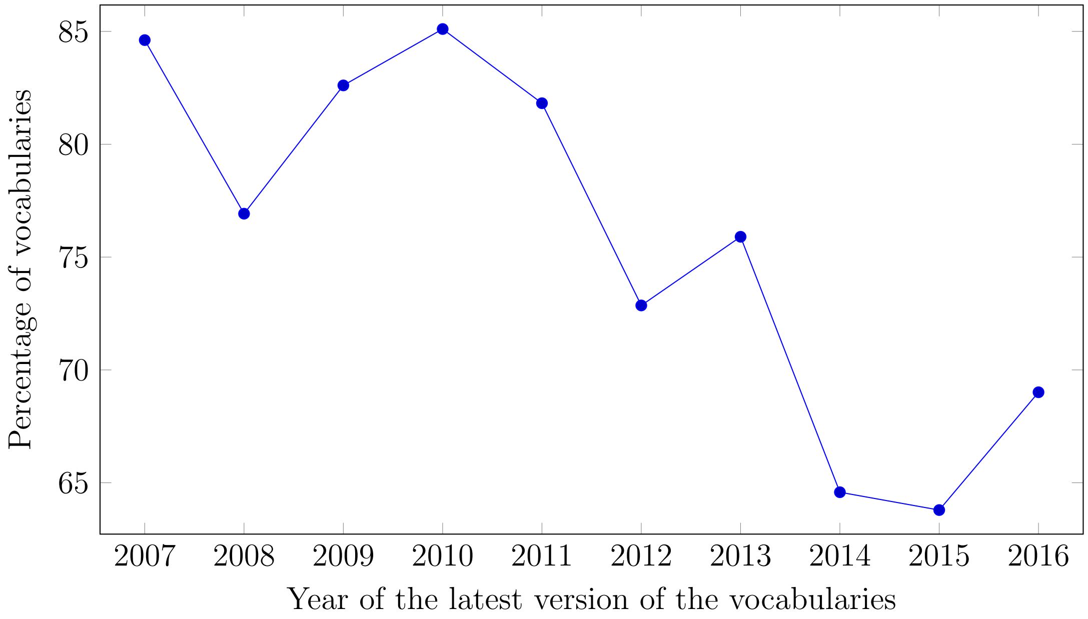


Figure 4-4: Distribution of the percentage of vocabularies with a difference in ranking due to the author based vocabulary ranking factor. Years with less than 10 vocabularies have been suppressed from the diagram as the data will skew the distribution.

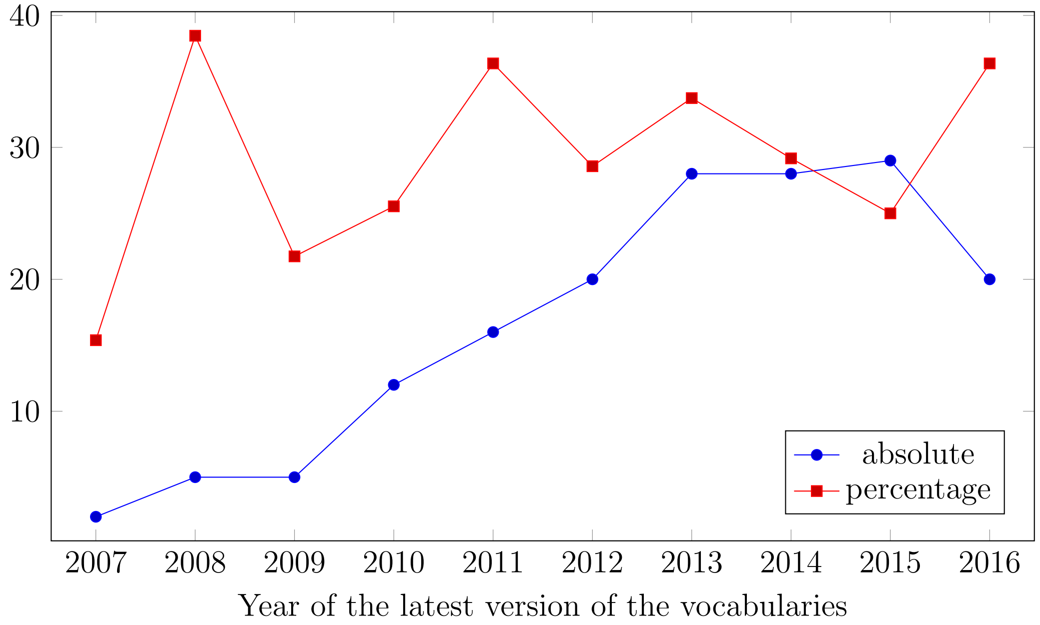


Figure 4-5: Distribution of the percentage of vocabularies and the absolute number of vocabularies with a difference in ranking due to the author based vocabulary ranking factor and zero starting score. Years with less than 10 vocabularies have been suppressed from the diagram as the data will skew the distribution.

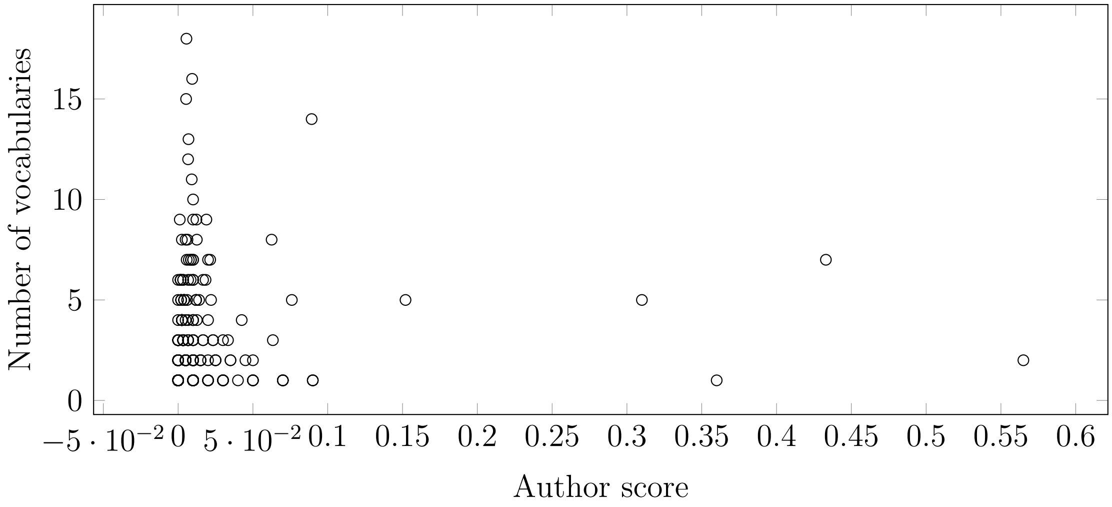


Figure 4-6: Scatter diagram of the LOV author scores and the number of contributed vocabularies. One outlier has been removed from the diagram in order to make it more readable. The outlier has value 1.130 and refers to one of the authors of the RDF recommendation, which is a vocabulary in the repository with the highest number of incoming links.

ommendation approach, the precision and recall metrics are defined as shown in Definition 18 and Definition 19, respectively.

Definition 18 (Vocabulary result terms precision) *If $V_{relevant}(w)$ represents the set of vocabulary terms that are relevant for a webpage w, $V_{retrieved}(w)$ is the number of vocabulary terms retrieved for w, then precision of the vocabulary terms recommendation is the fraction of retrieved vocabulary terms that are relevant to the webpage w:*

$$precision = \frac{V_{relevant}(w) \cap V_{retrieved}(w)}{V_{retrieved}(w)}$$

Definition 19 (Vocabulary result terms recall) *If $V_{relevant}(w)$ represents the set of vocabulary terms that are relevant for a webpage w, $V_{retrieved}(w)$ is the number of vocabulary terms retrieved for w, then recall of the vocabulary terms recommendation is the fraction of relevant vocabulary terms that have been retrieved for the webpage w:*

$$recall = \frac{V_{retrieved}(w) \cap V_{relevant}(w)}{V_{relevant}(w)}$$

The set of relevant terms for a given webpage w, i.e. $V_{relevant}(w)$, could be defined in many different ways. The first option is to generate an aggregated set of vocabulary terms from the manual annotations that were produced as part of the aforementioned survey. The second option is to employ a domain expert for the given webpage domain, who will be responsible to extract the core keywords from the webpage and later confirm the mappings of the keywords to manually selected terms from the vocabulary space. Finally, a third option is to leverage the domain expertise reflected in existing webpages that are semantically annotated by generating an aggregated vocabulary out of the terms that are used in a diverse set of similar webpages.

4.5 Vocabulary term search and ranking

Searching for vocabulary terms is similar to searching for vocabularies with major difference the aim of the search, which is the discovery of a term that better describes a given keyword. In addition, LOV search is capable of searching for vocabulary terms for a given keyword. The discovery of terms has also been the research subject for other approaches as it was presented earlier in the related work chapter (Chapter 3). One of them is the vocab.cc interface, described in Section 3.1.3, which aims to provide the best matches for a given keyword. The result vocabulary term URIs are accompanied with the number of occurrences and the overall score across the collection of documents that the search service is based.

This section defines ranking metrics that can be used in order to distill usage information for the Linked Open Data cloud regarding the various vocabulary terms of the vocabulary space. The focus of the approach is on two major datasets, namely the Billion Triples Challenge Dataset [37], [43] and the LOD cloud itself. The former is represented in the approach by the vocab.cc initiative, while the latter by the LODStats iniative [4].

Vocab.cc provides a few metrics for all the documented vocabulary terms, including the overall occurrences of a term, the number of documents that refer to a term, and the ranking metrics of the term in the corpus, i.e. the overall ranking and the document ranking. The ranking metrics provided by vocab.cc reflect the importance of the term among the documents in the corpus. An advantage of those metrics over the absolute figures is that they can be used in conjunction with other factors to produce a more comprehensive equation. As the figures are relative to the total amount of documents in the corpus, the metrics are normalised and prevent the generation of an unbalanced equation in case they are combined with other metrics, which are calculated on different datasets.

Therefore, the term ranking based on the Billion Triples Challenge Dataset using the overall ranking of a given term t and the document ranking is defined by Definition 20.

Definition 20 (Vocabulary term ranking in BTCD) *If $OR(t)$ is the overall ranking and $DR(t)$ the document ranking of a term t, $t \in v$ and $v \in \mathcal{V}$, then:*

$$TR_{BTCD}(t) = (OR(t) \cdot DR(t))^{1/2}$$

The formula of Definition 20 will return a lower value for those terms with a better ranking position within the dataset, thus the range of the metric is $TR_{BTCD}(t)$ $\in [1, +\infty)$. The geometric mean is preferred over the arithmetic mean as it gives a meaningful average between the overall ranking and the document ranking, while the arithmetic mean would not help to normalise the ranges. Thus, the arithmetic mean would allow the metric with the greater values to dominate the weighting between the two factors.

As it has already been mentioned in the related work, in Section 3.1.4, the LODStats initiative [4] is a great resource of statistical metrics about the LOD cloud. It contains 32 different metrics computed over stream based approaches, which enables its processes to handle millions of triples and scale up better than other approaches. Similarly to the BTCD dataset introduced within the scope of the previous metric, the integrated data from the LODStats into the LOV infrastructure is assumed to be adequate for the vocabularies and terms usage within the LOD cloud. This assumption is supported by the metadata that accompany the search results of vocabularies and terms, e.g. it is stated that the schema:Place occurs 3,164,782 times in 4 LOD datasets. Furthermore, browsing the metrics of various terms realised that the number of occurrences are distributed with great outliers. Thus, there are terms with millions of occurrences, e.g. schema:Place, while others are met a few thousand times, e.g. gr:Offering 22,584, or only a few times like schema:isRelatedTo, which has only 2 occurrences. However, for the context of the presented approach the usage of a term 17M times or 8M times does not make any significant difference about the popularity of the term, while it would make a difference if a term is used only once, 1000 times or 1M times. In this respect, the formula in definition 21 is based on the logarithmic function with base-2 as in principle logarithmic scales reduce wide-ranging quantities to smaller scopes. Thus, this formula will reflect better the profile of the previous mentioned data within a smaller range, i.e. 0 for $log_2(1)$ to ca. 24 for $log_2(17 \cdot 10^6)$. Base-2 is preferred over base-10 or ln due to the nature of the values that we deal with, causing the $\log_2$ results to spread more on the axis than the rest; allowing a better comparison of the calculated metric.

Definition 21 (Vocabulary term ranking in LOD) *If $OC(t)$ is the number of occurrences of a term t, $t \in v$, $v \in \mathcal{V}$, and $OC(t) \in [0, +\infty)$ then:*

$$TR_{LOD}(t) = \log_2(OC(t) + 1)$$

According to the aforementioned definition, the $TR_{LOD}(t)$ metric is higher for those terms that are more popular, with worst value to be the 0, which leads to the range of the metric to be: $TR_{LOD}(t) \in [0, +\infty)$. The formula adds 1 before computing the log base-2 in order to circumvent the issue of computing the ranking for a term with zero occurrences, which is not a real number. In addition, adding 1 to the occurrences does not change the impact of the final result.

As shown in Table 4.5, the foaf:Person term has significantly better score compared to schema:Person based on the $TR_{BTCD}(t)$, while they seem to be on par according

Term t	$TR_{BTCD}(t)$	$TR_{LOD}(t)$
schema:Place	45.72	21.59
dbpedia-owl:Place	46.73	5
foaf:Person	6.93	21.14
schema:Person	54.41	19.90
owl:sameAs	10.39	24.01

Table 4.5: LOD Term ranking examples (published in [79]).

to the $TR_{LOD}(t)$. The reason is that the former takes in consideration the number of datasets that a term occurs, while in the latter we include the occurrences decoupled from the number of documents that include them.

The aforementioned metrics combined with the vocabulary ranking metrics are used to build the core formulas of the proposed approach by leveraging all the quantitative dimensions of the various datasets related to vocabulary terms, including the Linked Open Vocabularies dataset, the Billion Triples Challenge Dataset and the LOD cloud via the LODStats collection of statistics. The application of the above defined ranking metrics on the presented approach is presented within the next chapter, which elaborates on the basic algorithm and the design of the methodology.

4.6 Summary

In this chapter, the theoretical base of the proposed methodology was presented together with the various definitions and the reasoning behind some basic design decisions. The chapter started by explaining the space in which the approach is meant to be positioned and also it briefly presented the ideas behind it. In brief, the major reason of this research direction, was the realisation of the overlapping between the website content and the API content that occurs in many of the popular websites and websites of various business domains. In this regard, the proposed approach aims to facilitate the generation of annotations for existing webpages in order to leverage them to machine understandable Web entities in the Linked Data space. For this to happen, searching for vocabularies and terms should be closer to the webpage developers and maintainers, helping them to choose vocabulary terms but also to educate them about the vocabulary space via the discovery funnel.

Therefore, the chapter continues by briefly presenting the various data sources of the recommendation base and by introducing the fundamentals for the vocabulary search topic and defines the metrics that drive the ranking of vocabularies within the vocabulary space. The definition of the ranking of vocabularies starts by demonstrating the first version of the PageRank algorithm, which was used in the Web scale and enabled the assignment of scores to webpages. Furthermore, the various definitions that lead to the proposed methodology are defined throughout Section 4.4. In addition, the chapter continues in Section 4.5 by introducing the ranking dimensions of the terms within the vocabularies, allowing to search and discover vocabulary terms for a given set of keywords. Finally, the discussion section provides a summarisation and overview of the approach aim and accomplishment while in parallel highlights any limitations that have been observed, but also any directions for further research on top of the presented work.

The next chapter, namely Chapter 5, dives in the application of the above presented theoretical basis by utilising the Linked Open Vocabularies (LOV) search endpoints. The architecture behind the methodology and the core algorithms that encapsulate the presented ranking metrics formulate the proposed methodology. The implementation of the methodology is later discussed in Chapter 6.

Chapter 5

Approach and Methodology design

The vocabulary terms recommendation framework

Combining the definitions and sources of data described throughout the previous chapters, the current one realises the theoretical base of Chapter 4 into the concrete framework that has been designed to propose vocabulary terms to a given set of keywords or to a webpage. Therefore, the main algorithm is explained within the sections that follow. The algorithm takes in consideration a few more dimensions on top of the ranking that the LOV directory and retrieval mechanism applies in order to sort the search results for the user. Thus, it can be considered as a LOV post-ranking algorithm to enrich the sorted results with even more dimensions. In addition, the presented approach acts as an aggregator by searching for a set of keywords to the LOV directory and providing a unified result set to the user. Also, it works as a metasearch service of vocabulary terms by being able to combine results from many sources of vocabularies and term usage directories, e.g. vocab.cc, LODStats, etc. Finally, reading through the design and implementation gives the impression that is built on top of the LOV search service, although it is agnostic to the service that provides the vocabulary terms candidate list. In this respect, the framework has been designed in order to be able to connect to different services as well by using a normalised interface that would be used to connect any other service than the LOV search.

The details of this chapter fall in the middle ground between the theoretical part and the practical implementation of the described framework. It mainly provides the methodology of the approach that has been based on the definitions and concepts of the previous chapter. Having described the methodology in this chapter, the reader will have the chance to dive into the implementation details in the next one, i.e. Chapter 6.

The chapter is organised in a way to help the reader understand the design of the approach by starting with the methodology definition in Section 5.1. The architectural description and the various components that compose the framework are presented in Section 5.2. After providing the bigger picture, it introduces the definition of the main algorithm that orchestrates all the steps in Section 5.3. In addition to the main algorithm that applies metrics on LOV and LOD to combine them via a scoring equation, one more dimension is described which is effective in a post-ranking stage and employs a set of patterns to detect predefined data types according to a knowledge base that has been designed in the scope of the approach. Section 5.4 provide additional dimensions in which the result vocabulary can be enriched beyond the recommendations of the algorithm. The subsections refer to the recommendation of vocabulary terms about the various multimedia elements of the webpage, about the recognition of specific datatypes based on predefined patterns and also discusses the actions dimension of the webpage entities. The actions of the annotated entities refer to potential ways to interact with the described entity. Furthermore, Section 5.5 introduces the *vSearch* vocabulary, which is used to describe the generated set of vocabulary terms together with the webpage keywords used as input. Finally, the chapter concludes with a summary and the outlook of the methodology (Section 5.6).

5.1 Methodology definition

The proposed approach can be framed within the category of design science research for information systems, as it fulfils the main characteristics of it, like the principle of solving observed problems by making research contributions and evaluating the results for their quality, utility and efficacy. Peffers et al. present in [62] a sound definition of the design science research in information systems by combining many resources from the theoretical base of the design science research approach. This paradigm provides great guidance to conduct the scheduled research by requiring the following six steps in order to accomplish the proof of the initial hypothesis.

1. *Problem identification and motivation.* At the beginning of the research process is the realisation of the problem that motivates the design of a solution. Earlier in Chapter 1, the main research questions together with the motivation of this work were presented as part of the introduction to the manuscript.

2. *Objectives definition for a solution.* The next step after having identified the problem is to define the objectives that are considered important in order to accomplish a solution to the previously described problem. In this scope, the main thesis of the research as it was formulated in Section 1.2 plays the role of the objective that will be used later during the evaluation to measure the effectiveness of the followed solution.

3. *Design and development.* The creation of the proposed solution is one of the core parts, including the research needed to design the methodology as well as the development of the approach to a tangible artifact. This chapter together with the next one, namely Chapter 6, present the design and development of the proposed approach.

4. *Demonstration.* Important part of the followed process is to demonstrate the application of the solution in use cases and explain the usage of it. In Chapter 6, the usage of the proposed methodology is presented while in Chapter 7 a few use cases are presented.

5. *Evaluation.* This part involves the comparison of the results that the designed solution provides with the objectives that were defined at the beginning of the research endeavours. Chapter 8 is dedicated to the effectiveness of the approach.

6. *Communication.* Research publications and presentations in conferences are responsible for the description of the problem and the proposed solution as well as to demonstrate the usage of it. In the scope of this PhD research work, there are both publications to describe the problem and the core parts of the solution ([79], [80]), but also publications to describe the design and development and demonstrate the usage of it [81].

The various steps are not necessarily addressed in the sequence that they appear in the list, but some of them could happen in parallel, like the *Communication.* Very often early results are published and presented in conferences before all the previous steps have been completed. This was the case for the presented research as the various contributions were presented before the whole work was completed.

5.2 Discovery of vocabulary terms

The discovery of vocabulary terms as a process depends first and foremost on the input content that needs to be annotated. At the search layer, which refers to the search of terms in the vocabulary space there are two main aspects that significantly steer the result vocabulary, namely the ranking of the vocabularies and the ranking of the term as part of the vocabulary space and as part of the Linked Open Data cloud. The ranking of vocabularies and terms were discussed in Sections 4.4 and 4.5, respectively. Combining the metrics and dimensions of the two aforementioned aspects, this section introduces new metrics that are used in the discovery process, as defined in Definition 22 and later in the main algorithm of the approach, namely the LOVR algorithm, which is presented in Algorithm 1 of Section 5.3.

The aim of the approach is to generate suggestions about the best candidate terms for the implementation of semantic annotations on a webpage based on the two aforementioned aspects. Those two parameters define the overall ranking of the term, namely the Linked Open Term Rank (LOTR) as presented in Definition 22. The $LOTR$ ranking metric aims to assign a score to each single term t of the extracted vocabulary terms' set by accounting the importance of the vocabulary across the LOV space as demonstrated in Section 4.4; the usage of the term by the existing LOD datasets; and the relevance of the term to the related keyword of the input content or webpage. The range of $LOTR$ is the set of positive real numbers including zero, $LOTR \in \mathbb{R}_{\geq 0}$ and a higher score denotes a higher position in the ranking sequence.

Definition 22 (Vocabulary term ranking) *If $re_{t,k}$ is the relevance of the term t to a given keyword k, $re_{t,k} \in$ (0,1], $VSR_{LOV,V}$ is the ranking of the vocabulary V within LOV, $VSR_{LOV,V} \in (0,1]$, $TR_{BTCD,t}$ is the ranking of t in the BTCD and $TR_{LOD,t}$ is the ranking of t in the Linked Open Data cloud based on its usage, and α is a constant factor then:*

$$LOTR_t(k) = (\frac{TR_{LOD,t}}{TR_{BTCD,t}} + \alpha \cdot VSR_{LOV,V}) \cdot re_{t,k}$$

The $TR_{BTCD,t}$ factor of the formula is at the denominator of the fraction due to the range of it, as described in Definition 22, and the fact that a smaller value represents a better ranking position of the term in BTCD. In addition, the α constant has been introduced in the formula in order to address the problem of having the multiplication of two factors with values below 1, i.e. $VSR_{LOV,V}$ and $re_{t,k}$, which would have a product lower than each of the factors. It can be assumed that $\alpha = 100$, which will be enough in order to overcome the multiplication issue, but from a mathematical angle, any value of this magnitude would suffice and would not alter the results and the ranking of the terms. The relevance of the given keyword k to the vocabulary term t, i.e. $re_{t,k}$, reflects the string similarity of the two string (keyword and term). The distance between them can be calculated using any of the string distance algorithms that are widely used, e.g. the Jaro-Winkler distance, the Levenshtein distance, etc., and it should be normalised, such as $re_{t,k} \in$ (0,1].

Table 5.1 depicts the LOTR scores for various candidate terms about various keywords. For the first keyword, i.e. *Place*, according to the assigned scores, the *schema:Place* and *event:Place* are the highest ranked terms while the former wins

Term t	Keyword k	$re_{t,k}$	$LOTR_{t,k}$
schema:Place	place	0.784	2.94
event:Place		0.751	0.21
dbpedia-owl:Place		0.458	0.05

Table 5.1: LOTR concept ranking example.

Metrics		**Description**
Approach	**LOV**	
$VSR(v)$	LOV tf-idf	Vocabulary ranking.
$TR_{BTCD}(t)$, $TR_{LOD}(t)$	$pop(t, D)$	Term popularity score.
$re_{t,k}$	$norm(t, v)$	Similarity of the keyword to terms.
$LOTR_t(k)$	$score(t, k)$	The term's ranking score for a given query.

Table 5.2: Comparison of the metrics defined in the proposed approach and LOV according to [90].

over the latter mainly due to the number of occurrences in the LOD datasets as the LOVRank $VR_{LOV,V}$ scores are very close to each other.

Moreover, after studying the metrics that are used in the various LOV services, as presented in [90], the above defined metrics of the proposed approach can be modified in order to function as post-ranking metrics on top of the LOV metrics, in case that the LOV services are used at the base of the vocabulary terms discovery. In this regard, Table 5.2 compares the metrics of the proposed approach with the metrics that are available in the LOV metrics and shape the ranking of the terms. As the comparison shows, the presented approach proposes for the same type of metrics different formulas to calculate the score. Therefore, it would be possible to substitute the LOV scores, but also it would be possible to apply the presented metrics in a post-ranking fashion. In this regard, the relevance score $re_{t,k}$ in the $LOTR$ formula could play the role of the LOV ranking score, which encapsulates the similarity but also all the rest of metrics.

5.3 The vocabulary terms recommendation algorithm (LOVR)

Main purpose of this section is to analyse the methodology that combines all the aforementioned theoretical parts and formulas under the scope of the LOVR framework and the main algorithm that resides at the core of it. Earlier in Figure 4-2, the workflow diagram depicts from a higher level the input and output of the methodology that has been designed. The input of the methodology is the target webpage and the output is a set of vocabulary terms from the vocabulary space, which forms the *result vocabulary* $\mathcal{T}$. The framework consists of all the necessary modules to discover vocabulary terms for a given webpage (URL), as shown in Figure 5-1.

The *LOVR* module is considered the core of the framework and the one that orchestrates the data flow in the methodology. This core module is communicating with all the search related modules that are handling the search within the Linked Open Data cloud and the vocabulary space, respectively the *LOD Search* and the *Vocab Search* modules as shown in the components diagram.

The *Natural Language Processing (NLP)* component, that is depicted in the diagram, is required in order to process the content of the webpage and extract keywords based on the selection of the most important tokens of the content. The extracted keywords are used as input to the other two core components to find related terms within the vocabulary space. Moreover, the NLP component refers mostly to the implementation of the approach, which is discussed later in Chapter 6, and it can be considered optional in the scope of the methodology. The alternative configuration receives at the input a set of keywords, instead of relying on the extraction process of the NLP component. In this scenario, the diagram of Figure 5-1, will not have as input the shown URL, but a set of keywords, which the LOVR component will consume without the need of going through the NLP component.

One of the important assets of the methodology that differentiates it from the rest is the ranking metrics based on the vocabulary space and on the LOD cloud. The various ranking scores are utilised either within the LOVR component or the two components that are related to the search of terms and vocabularies. This is an implementation detail, which is not studied within this chapter as it is considered unrelated to the methodology design. However, the algorithm that combines all the metrics and computes the final ranking score for each one of the retrieved terms is part of the LOVR component that is depicted in the conceptual architecture diagram (components diagram).

Finally, the *LOD Search* and the *Vocab Search* components are responsible for the communication of any external services related to them, like the *vocab.cc*, the *LODStats* and the Linked Open Vocabularies (*LOV*) directory of vocabularies.

The *LOVR* algorithm, that resides at the core of the approach, can be described as an iterative process that starts from the extraction of the keywords k from a webpage $\mathcal{W} = \{k_1,\ k_2,\ ...,\ k_n\}$ and then iterates through all of them. The aim is to find the best matching vocabulary term t for each keyword k. For each one of the extracted keywords a search in the LOV directory is performed in order to find the most relevant terms, which form the set $V_{LOV,k}$. All those terms are accompanied with a relevance score re, provided by the LOV search. Therefore the search result set, can be defined as:

$$V_{LOV,k} = [t_1, re_1], [t_2, re_2], ..., [t_n, re_n], n \in \mathbb{N} \tag{5.1}$$

Thus, for each keyword k there is a set $V_{LOV,k}$ with those vocabulary terms that best match based on the score that LOV search materialises (including text similarity). For each term t_n of this set, the algorithm executes the computations for the Linked Open Term Rank $LOTR_k(t)$ formula as described in Definition 22. The result is kept together with the results for the rest of the terms of the $V_{LOV,k}$ set and then the term with the highest score is marked as the most appropriate match for the given keyword k. Putting together the best candidate term t with the keyword k, a pair is created, which is one of the items of the set $\mathcal{T}$ that will be returned as output of the algorithm. In addition to the pair, for transparency and deeper understanding of the approach, the ranking score $r_{k,t}$ is stored together with the pair by formulating a

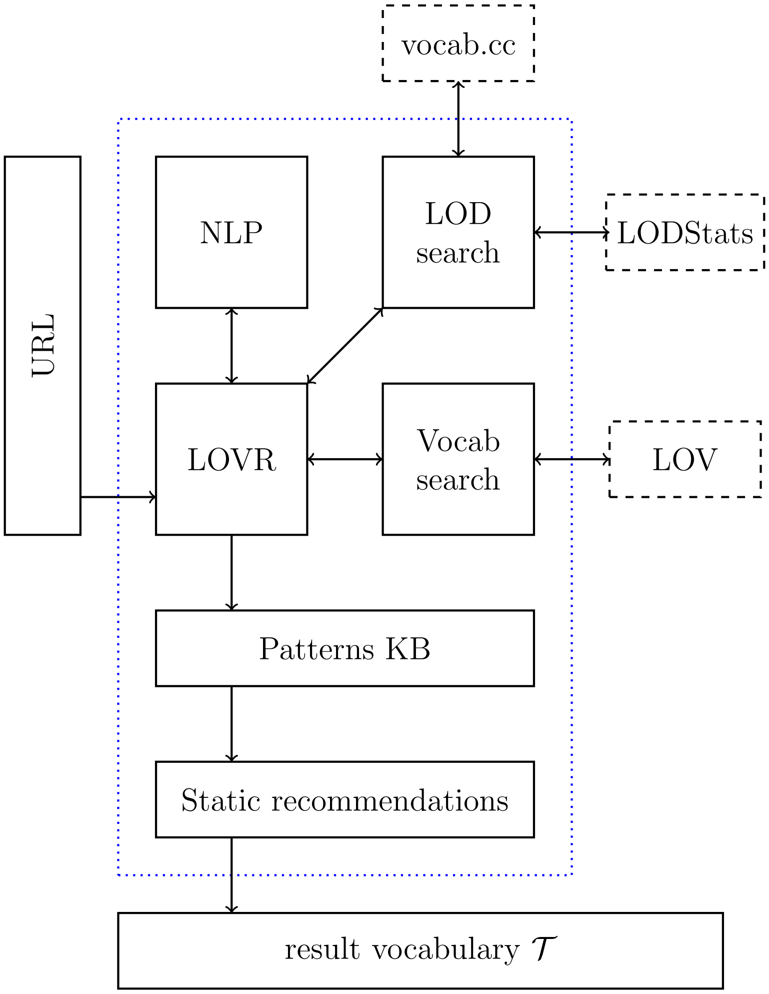


Figure 5-1: The LOVR methodology conceptual architecture diagram with the involved components. The modules enclosed within the dotted line represent the methodology, namely the NLP (Natural Language Processing), the LOD Search, the Vocab Search, the LOVR algorithm, the Patterns Knowledge Base and the Static recommendations. The rest of the elements refer to external services that are integrated (vocab.cc, LODStats, LOV). A more detailed architectural diagram is depicted later in Figure 6-2.

Variable	Definition
$\mathcal{W}$	Set of keywords for a given webpage.
k	Denotes a keyword.
t	Denotes a vocabulary term.
$LOTR_k(t)$	Linked Open Term ranking of term t for the keyword k.
$V_{LOV,k}$	Set of vocabulary terms returned by LOV Search for the keyword k.
$r_{k,t}$	Ranking score of term t for the keyword k.
R_k	Set of ranking triples for the keyword k.
$\mathcal{T}$	Set of terms that compose the *result vocabulary*.

Table 5.3: Variables used throughout the LOVR algorithm.

ranking triple, as defined in Definition 23.

Definition 23 (Ranking triple) *Ranking triple (k, t, r) is defined as a sequence of values that connects a given keyword k with a vocabulary term t and the corresponding ranking score r for k over t in a given vocabulary space.*

Furthermore, another version of the algorithm could return a list of ranking triples for a given keyword k, instead of one single result. This would make the result vocabulary $\mathcal{T}$ to grow in size, which would add complexity, but at the same time it would allow to expose candidate terms together with their ranking scores and leave the decision to the user of the methodology about which one would be used.

As shown in Table 5.3, which defines the various used variables in the algorithm presented in Algorithm 1, $\mathcal{T}$ is a set of ranking triples (keyword, term, ranking). The goal of the LOVR algorithm is to return a set $\mathcal{T}$ for a given webpage.

At the core of the algorithm stands the LOTR formula which is applied to every term that has been correlated with the keywords of a webpage. Therefore, the LOTR scores are those that steer the decisions within the algorithm and support the formation of the $\mathcal{T}$ set. The user of the methodology is given a more flexible answer by including in the result set not only the top performing terms as long as the ranking score is concerned, but also a few candidates that could better address the needs of the user compared to the top performing term.

Beyond the aforementioned graph based ranking of vocabularies and suggestion of terms based on the HTML elements semantics, one more dimension is discussed

```
Data: Webpage content
Result: Set of vocabulary terms to annotate the given content
extract the keywords of the webpage: $\mathcal{W}$={$k_1$ ... $k_n$};
foreach keyword $k \in \mathcal{W}$ do
    /* $V_{LOV,k}$ keeps a set of terms from LOV                    */
    $V_{LOV,k} \leftarrow$ LOVSearch($k$);
    foreach term $t \in V_{LOV,k}$ do
        $r_{k,t} \leftarrow LOTR_k(t)$ ;
        /* $r_{k,t}$: ranking of $t$ for $k$                        */
        add $r_{k,t}$ to the $R_k$ set ;
    end
    $r_{max} \leftarrow$ max($R_k$);
    add the pair $k, t$ that corresponds to the $r_{max}$ to the $\mathcal{T}$ set;
end
return $\mathcal{T}$;
```

Algorithm 1: The LOVR algorithm.

below, i.e. the feedback loop or training phase. In general, it is considered crucial to incorporate a relevance feedback loop in the information retrieval systems. In the case of the presented approach the relevance feedback loop can be considered as a continuous training phase for supervised learning to rank. The aim of this part of the approach is to provide a score about the relevance of the results that can be used to shape the final ranking of the result terms. In this respect, the user is giving feedback about the vocabulary terms suggestions, which is stored appropriately in order to be incorporated in the future suggestions. Definition 24 describes the feedback representation within the methodology when the feedback loop is implemented.

Definition 24 (Feedback vector) *Let $\vec{F}$ be the feedback vector of the term t to a given keyword k, $LOTR_t(k) \in$ (0,1] the term ranking in LOVR, $rel \in$ {0,1} the relevance score, and d the domain of the document, then:*

$$\vec{F} = \{k, t, LOTR_t(k), d, rel\}$$

Therefore, the approach includes a learning procedure to assist the ranking algorithm with knowledge acquired from previous observations. The more observations and usages the framework has, the better results we expect it to provide and be able to adapt and evolve in time as the Web evolves in parallel, too.

5.3.1 Pattern based vocabulary terms suggestion

In addition to the above-described methodology that is based on a keyword search against the vocabulary space, one more source of recommendations is considered as an important asset of the core. This part refers to the extraction of various datatypes that appear in the content of the webpage. The LOVR algorith (Algorithm 1)has a drawback related to datatype names that do not appear witin the text of the webpage, but they are represented only by the value of the corresponding instance. An example is the representation of an email address, which would appear within the webpage as a literal of the format *name@someorganisation.com* but the word *email* could be neglected as for a human user of the webpage it is obvious what the value represents to.

As it was already said, values of instances for classes like an email address and a phone number are not easily extracted by automatic approaches, as the name of them does not appear within the content in order to be later searched in the LOV space. In this respect, the presented methodology employs one more layer in the vocabulary terms recommendation which is based on the exploitation of predefined patterns that are able to map specific content parts to data types (classes), which are further mapped to specific vocabulary terms. The set of vocabulary terms generated based on patterns recognition within the webpage content is used to enrich the existing result vocabulary by combining the two as a union set.

The set of patterns presented in Table 5.4 works as a knowledge base on data formats and can be easily expanded to include as many as can be constructed. With the current ensures that no collision can happen in terms of mapping a data type to more than one vocabulary terms, as each format is mapped with a vocabulary term using an one-to-one relationship. The patterns for persons and organizations are inspired by the work in OnTeA [49]. The set of formats presented in the table

<table>
<tr><th>Type</th><th>Term</th><th>Pattern (Regular expression)</th></tr>
<tr><td>email</td><td>schema:email</td><td>[\S]+@[\S]+</td></tr>
<tr><td>person</td><td>schema:Person</td><td>(Mr.|Mrs.|Dr.|Prof.)\s([A-Z][a-z]+\s[A-Z][a-z]+)</td></tr>
<tr><td>company</td><td>schema:Organization</td><td>([A-Za-z0-9]+)[,\s]+(Inc|Ltd|GmbH)</td></tr>
<tr><td>phone</td><td>schema:telephone</td><td>\+\d{1,4}?[-.\s]?\(?\d{1,3}?\)?[-.\s]?\d{1,4}[-.\s]?\d{1,4}[-.\s]?\d{1,9}</td></tr>
<tr><td>time</td><td>schema:Time</td><td>([0-9]|0[0-9]|1[0-9]|2[0-3]):[0-5][0-9]</td></tr>
<tr><td>date</td><td>schema:Date</td><td>(\d{1,2}(-)\d{1,2}(-)\d{4})|(\d{4}(-)\d{1,2}(-)\d{1,2})</td></tr>
<tr><td>duration</td><td>schema:Duration</td><td>\d+\s(min|hour|hr)</td></tr>
<tr><td>price</td><td>schema:price</td><td>\p{Sc}\s?\d+|\d+\s?\p{Sc}</td></tr>
</table>

Table 5.4: The set of data type patterns used to discover vocabulary terms (Patterns Knowledge Base).

is only a subset that can be included, and showcases the idea behind the inclusion of regular expressions to define vocabulary terms for a given webpage or document. In addition, it is important to understand that the regular expressions are mainly used to decide if a format is met with in the document or not and not to extract the text of the document. Therefore, the expressions can be more relaxed than trying to validate and precisely extract text from the content. For example, the *duration* type of Table 5.4, matches when there is a number with any amount of digits followed by any whitespace character and the literals *hour* or *minute*, neglecting the plural version of the words. Table 5.4 consists of three parts. The first column refers to the data type that the pattern aims to locate within the webpage, the second column reflects the vocabulary term that it is mapped to, while the third column presents the regular expression that is represents the pattern and is used to detect the mapped data type in the webpage. All of the terms that have been proposed, as mappings to the data types of the *Pattern KB*, stem from the schema.org vocabulary as it has very rich and abstract models that should cover the needs of any webpage.

Listing 5.1 depicts an example of a document showcasing how the *Pattern KB* enables the approach to detect types that otherwise would have been missed via the token based keyword search of terms within the vocabulary space. The textual content of Listing 5.1, refers to a recipe by a fictional author. The details about the recipe include the needed time in total and the time required by the cook to prepare and follow the instructions. However, there is not any word in the text that would allow the extraction process to realise that the amount of time refers to duration. In this case, the *Patterns KB* facilitates the recognition of the values *4 hrs* and *30 minutes* that represent some duration within the recipe. In addition, the *person*, the *email* and the *telephone* data types are able to be extracted from the second half of the content. Therefore, the pattern based generated vocabulary would be: $V_P = \{$schema:email, schema:Person, schema:telephone, schema:Time, schema:Duration$\}$.

```
From a pizza recipe:
Total: 4 hrs
Active: 30 minutes
Yield: 1 pizza, serves: 2 (2 servings per pizza)
...
Author: Mr. Nick Chef, nick.chef@gmail.com, +49150111010
```

Listing 5.1: Example text that would benefit by applying the Pattern KB to the content.

Figure 5-1 shows the *Pattern KB* module of the proposed methodology and its placement in the conceptual architecture of the approach. As the diagram reflects, the application of data type patterns on the target webpage takes place after the LOVR algorithm has finished. Therefore, any terms that are extracted via the pattern set and do not appear in the LOVR result set are being added to it. The set of patterns is applied against the content in an iterative fashion as described in Algorithm 2. Therefore, at the end of the iteration, a new set of vocabulary terms will have been created, which will reflect the patterns that were successfully matched against the content.

Algorithm 2 reflects the post-LOVR discovery, which is based on data type patterns. The algorithm is based on the application of the patterns from the pattern knowledge base. The patterns are applied one after the other and if there is a match, then the corresponding vocabulary term is added to the set of vocabulary terms V_P

```
Data: Webpage content, $\mathcal{T}$ vocabulary
Result: Set of vocabulary terms to annotate the given content
Let $\mathcal{P}$ be the set of patterns in the Knowledge Base: $\mathcal{P}$={$p_1$ ... $p_n$} ;
foreach pattern $p \in \mathcal{P}$ do
    /* Let $V_P$ be a set of terms from patterns extraction */
    /* Let $t_p$ be the corresponding term to pattern $p$ */
    if $p$ matches website content then
        add $t_p$ to $V_P$;
    end
end
$\mathcal{T} \leftarrow \mathcal{T} \cup V_P$ ;
return $\mathcal{T}$ ;
```

Algorithm 2: The extended LOVR algorithm with pattern based extraction and static enrichment.

that stem from this process. Afterwards, the union of V_P with the result vocabulary $\mathcal{T}$ is returned as the new and enriched result vocabulary. According to the set theory, a set cannot have duplicates. Thus, it is not necessary to compute the intersection and then the union in order to avoid having duplicate terms in the result vocabulary.

5.4 Result vocabulary enrichment

The discovery of terms, as presented in the previous sections, handles the search of keywords against the various directories of vocabularies and Linked Open Data in order to extract those vocabulary terms that better match the needs of the webpage that the keywords were extracted from. In this direction, the vocabulary generation complements the discovery effort by placing itself on top of the results that the search and ranking produced. The *result vocabulary* $\mathcal{T}$ has been computed throughout the previous sections as a result of the methodology and algorithm described in Section 5.3 and depicted in Figure 5-1 Algorithm 1, respectively. The $\mathcal{T}$ vocabulary, following the above described algorithm, includes one to three matches for each keyword (ranked based on the calculated score), but on the other hand it is missing vocabulary terms that could be used to annotate parts of the webpage that are not related to the textual content, which was analysed in order to prepare the *result vocabulary*.

In this scope, Section 5.4.1 discusses the discovery and inclusion of static vocabulary terms in the *result vocabulary* as those are defined by Definition 25. Furthermore, the suggestion of action vocabulary terms is studied in Section 5.4.2, which facilitate the interaction of the annotated entities of the webpage with the Web services world.

5.4.1 Static vocabulary terms suggestion

Throughout the previous section, we saw how the various terms are ranked in order to produce the set of suggestions. However, examining the produced set of terms from the aforementioned approach, we realise that a whole category of vocabulary terms is missing from the result vocabulary. This category refers to terms that could be considered as "static", because they are not coupled to the content of the webpage, but are more generic. In Definition 25, the static vocabulary term is defined as a term that derives from the structural elements of a webpage rather than from the content of it in terms of textual representation. For example, the description of various media types, like images is oblivious to the rest of the webpage content. Therefore, this special category is based on structural components of the webpage, e.g. specific HTML elements.

Definition 25 (Static vocabulary term) ***Static vocabulary term*** *is considered a vocabulary term t that describes a structural part of a webpage w and is unrelated to the textual content of w.*

An explicit set of rules is sufficient to address the recommendation for this category of terms. In the scope of the proposed approach the various elements are mapped to vocabulary terms in a static way, not allowing to dynamically map them to any possible new terms that could be introduced in the future in the vocabulary space. However, according to the best practices in ontology design, the future vocabulary engineer should reuse the existing terms instead of introducing new equivalent terms, which makes the static mappings approach not limited as soon as there are terms that can soundly address the needs. There are already many terms in very popular and widely accepted vocabularies that cover the needs of media content annotations with

HTML element	**Term t**	**Term t range**
<img>	schema:image	schema:URL or schema:ImageObject
<video>	schema:video	schema:VideoObject
<audio>	schema:audio	schema:AudioObject
<h1>	schema:name	schema:Text
<a>	schema:url	schema:URL

Table 5.5: Mappings between multimedia HTML elements and vocabulary terms used by the recommendation algorithm at the second recommendation stage. The *schema:* namespace stands for the URI `http://schema.org/`. This table has been published in [80].

the best one to be the *schema.org* vocabulary terms due to the extensive property set that is connected to the the corresponding terms.

In Table 5.5, we can see the used mappings by the static parts recommendation. It mainly includes mappings for the various media types to vocabulary terms from the schema.org vocabulary. The schema.org terms for image, video, audio provide clear semantics with broad domain and specific range, which can be reused in any context. On the other hand, there are many vocabularies that provide terms for the description of the various media types, but they target specific use cases and reusing them could lead to wrongly applied semantics in a webpage or document, or the previously mentioned concept of "vocabulary terms hijacking". For example, the result list of the "image" LOV search includes the *ebucore:hasRelatedImage*, which is part of the EBU (European Broadcasting Union) Ontology, which refers to a specific *BusinessObject* defined in the ontology and does not function as a generic type that could be used beyond the domain of the corresponding ontology.

In regard to the selection of the appropriate vocabulary term for the description of an image, a vocabulary engineer could reuse the term defined in a broadly accepted vocabulary other than *schema.org*, the *FOAF*. However, as it is mentioned in the vocabulary description[1] the *foaf:Image* term is equivalent to the schema.org term, recognising the need of having one canonical vocabulary in the vocabulary space. Furthermore, the first result in LOV for the search term *image*, appears to

[1] FOAF changes 2014: `http://xmlns.com/foaf/spec/#sec-changes20140114`

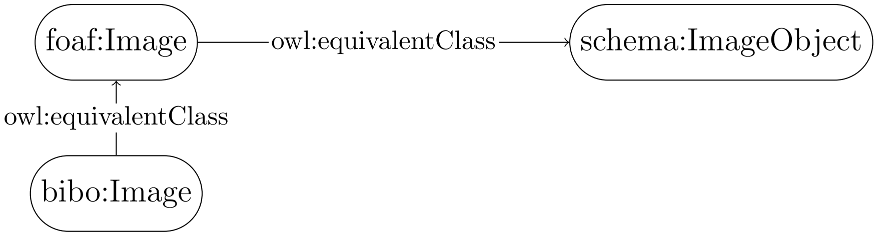


Figure 5-2: Relationships between various terms about an image object in different vocabularies, i.e. schema, foaf and bibo.

be the *bibo:Image* out of the 826 results[2], which is defined in the bibo vocabulary[3] as an equivalent to the *foaf:Image*. Also, it refers to an image of an instance of *bibo:Document*, which is defined as an equivalent to the *foaf:Document*, which is defined in FOAF as an equivalent of the *schema:CreativeWork*, following again the need of referring to a common canonical vocabulary. The above described relationships are depicted in Figure 5-2.

Studying the *schema.org/ImageObject* shows the comprehensiveness of the set of terms as long as the description of an image media type in the scope of a webpage is concerned. The properties of the *ImageObject*[4] include the following: *caption* for the caption of the object, *exifData* for exif metadata of the object (mostly relevant to photography related webpages), *representativeofPage* flag that explicitly allows to indicate if the image is considered representative to the webpage, *thumbnail* to link the given *ImageObject* to a different *ImageObject* that could play the role of a thumbnail, *associatedArticle* used to associate the image to a *NewsArticle* (mostly relevant to news portals), *author or creator* that refer to the author of the image, *contributor* for a secondary "author", *license* that refers to the license that applies to the image, and many more properties that can describe the *ImageObject* in even more details. The rest of the media types of Table 5.5 can be described in similarly extensive set of properties.

Together with the suggested static terms a few more are considered at the basic subset that the user of the methodology should employ for the semantic annotations

[2] LOV search for image: `http://lov.okfn.org/dataset/lov/terms?q=image&page=1`

[3] The Bibliographic Ontology (bibo): `http://purl.org/ontology/bibo/`

[4] Schema.org ImageObject description: `http://schema.org/ImageObject`

Term property t_p	**Media type**	**Term property purpose**
caption	image, video	The caption of the media object
representativeOfPage	image, video	Flag that explicitly allows to indicate if the image is considered representative to the webpage
creator	all	The creator of the media object
associatedArticle	all	Used to associate the media object to a *NewsArticle*, when that is relevant
license	all	The license that applies to the media object
copyrightHolder	all	The party holding the legal copyright to the media object
contentUrl	all	The actual media file that is described

Table 5.6: Property terms for the description of details related to the main media types.

generation. This subset is described in Table 5.6 and provides only a few of those that can be used in addition to the main vocabulary terms.

5.4.2 Action vocabulary terms suggestion

As it was presented in [78], the idea to build web agents to understand Web content and interact with it in order to realise a given plan and achieve goals has been part of the Semantic Web vision [28] since the very first steps of the related working groups. As described by Hendler in [39] with the example of intelligent travel agents, these systems should be *communicative, capable, autonomous* and *adaptive*. The various vocabularies and ontologies, that have been developed until today, contribute towards making Web content machine-understandable through semantic representation and annotations, which will enable Web agents to behave as described before. Since the version of schema.org introduced in 2014 [12], a new dimension has been added to the vocabulary which is related to the description of actions[5] that can be used to interact with the entities of the webpage. It has been designed with the flexibility of combining any *Action* to any type of object (class instance) of the vocabulary. This is allowed by design as the related property has been added to the most generic class, the *Thing*. Specifically, the action property is named *potentialAction* with domain the

[5] Schema.org actions documentation: `https://schema.org/docs/actions.html`

class *Thing* and range the class *Action*, which is at the top of the inheritance hierarchy of the actions. Thinking the actions in triples representation, the schema has been designed in such a way that more than one triple could be created in order to enable a fine-grained description of the action, participating agents, objects, results of it and time variables. A *triple (s, p, o)* as it is defined in the W3C RDF recommendation[6], contains three parts, namely the subject, the predicate and the object.

In this direction, this section introduces the generation of recommendations about potential actions, which the semantically annotated Web entities of a webpage could support. This part of the approach is based on the following assumption: *Let w be a webpage which has been semantically annotated and contains entities that would be meaningful to interact with other entities in the Web sphere.* For example, in case of lodging businesses, the offered services (e.g. reserve rooms) could be potential objects that an agent would be interested to interact beyond a simple read-only relationship (e.g. to make, update or cancel a reservation).

Definition 26 (Action properties) ***Action properties*** *is the set of object or datatype properties P_A whose subject can be a class that is either the schema:Action class or a subclass of it. The schema: namespace refers to the schema.org vocabulary.*

Definition 27 (Class properties) ***Class properties*** *is the set of object or datatype properties P_T whose subject can be any other class than the schema:Action class or a subclass of it. The schema: namespace refers to the schema.org vocabulary.*

According to Definition 26 and Definition 27 the intersection of the two property sets is the empty set, $P_A \cap P_T = \varnothing$. This artificial distinction between the class properties helps to better describe the approach that follows. Let $W = \{E_1, E_2, \ldots, E_n\}$ with $n \in \mathbb{N}$ be the set of all entities described in a website. Each entity $E \in W$ is the subject for the set of properties $P_T \cup P_A$.

[6] W3C RDF recommendation: `https://www.w3.org/TR/2004/REC-rdf-concepts-20040210/#section-triples`

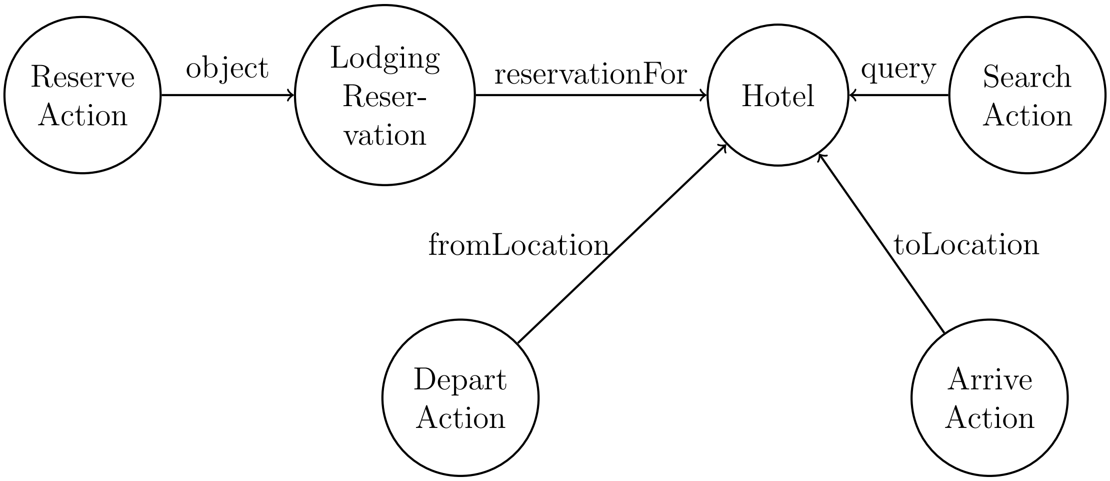


Figure 5-3: Hotel class interaction with schema.org actions (published in [78]).

Let $T = \{T_1, T_2, \ldots, T_t\}$ be the set of all entity classes in schema.org and $A = \{A_1, A_2, \ldots, A_m\}$ with $t, m \in \mathbb{N}$ be the set of all action classes in schema.org. Then, function ϕ maps each entity to the set of actions that could be used by the entity type:

$$\phi : T \to \mathcal{P}(A) \tag{5.2}$$

Therefore, having specified the mappings between classes and the possible action classes by utilising the equation of 5.2, it is feasible to define an entity E and its properties P_T and P_A. The presented approach aims to define potential actions for a class T, by studying the range of the properties in an *Action* class A. In this regard, two possible cases have been identified. The first case, which is the simplest one, refers to the direct mapping of the range of P_A to classes T. For example, according to Figure 5-3, the *DepartAction* $A_1 \in A$ has the property *fromLocation* $P_{A1} \in P_A$ with range *Place* $T_1 \in T$. Thus, any subclass of T_1 is in the range of P_{A1} and we could assume the A_1 as a potential action for the class *Hotel* (Place → Hotel).

The second case assumes that when properties P_A of a class $A_1 \in A$ and properties P_T of a class $T_1 \in T$, and P_T are not derived from superclasses (e.g. the *Thing*), have matching ranges, then A_1 is a potential action for T_1. This approach is applied to the example of Figure 5-3 as shown in Table 5.4.2. In particular, the *LodgingReservation*

LodgingReservation		maps to	ReserveAction	
property	type		type	property
checkinTime	*DateTime*	→	*DateTime*	scheduleTime
checkoutTime	*DateTime*	→		
lodgingUnitD.	*Text*			
lodgingUnitT.	*Text*			
numAdults	*Number*			
numChildren	*Number*			

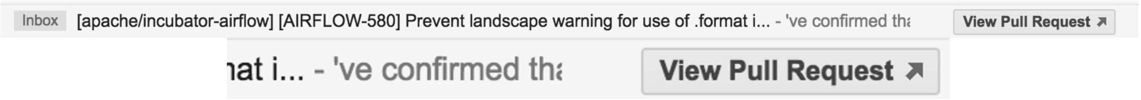
Table 5.7: Mapping the *LodgingReservation* to the *ReserveAction* (published in [78]).

class and the *ReserveAction* of the *schema.org* vocabulary are described in the table with all their properties, excluding any derived properties from superclasses. The former models a reservation entity of a lodging business and the latter depicts an action of reserving concrete objects like a hotel or a restaurant. The middle column of the table indicates the fields that map from the *Reservation* class to the related *Action* class. As it is shown, the range of all the properties of the *Action* class map to a subset of the *Reservation*'s properties.

The uptake of the actions has not been researched by the related literature, and as it is a new concept only a few examples can be found that the action classes and properties have been employed. The most popular example is the Gmail client actions as described in the corresponding documentation[7]. Gmail supports a) the *schema:RsvpAction* as that described in the corresponding schema.org documentation page[8], which can be used for informing event organisers about the attendance of the user; b) the *schema:ReviewAction* to express a review for restaurants, movies, or other products and services in general; c) the *schema:ConfirmAction* to approve or acknowledge something by one click; d) the *schema:ViewAction* for content consumption related links; e) the *schema:TrackAction* for any links that are related to parcel or any other post tracking process and the *SaveAction*, as it is called within the Gmail documentation (missing from the schema.org description) that can be used to describe URLs that the user would follow to save electronic books or other digital material.

[7] Gmail actions: `https://developers.google.com/gmail/markup/overview`

[8] Schema.org RSVP action: `https://schema.org/RsvpAction`



Inbox [apache/incubator-airflow] [AIRFLOW-580] Prevent landscape warning for use of .format i... - 've confirmed thɛ

Figure 5-4: Gmail action button as appears in the Web browser based client at the inbox list of emails. The example refers to a GitHub repository notification email about a pull request.

```
<div itemscope itemtype="http://schema.org/EmailMessage">
<div itemprop="action" itemscope
    itemtype="http://schema.org/ViewAction">
  <link itemprop="url"
    href="https://github.com/apache/incubator-airflow/pull
       /1843"/>
  <meta itemprop="name" content="View Pull Request"/>
</div>
```

Listing 5.2: Example of a *schema:ViewAction* used in emails by GitHub using Microdata.

```
<script type="application/ld+json">
{  "@context":"http://schema.org",
   "@type":"EmailMessage",
   "action":{
         "@type":"ViewAction",
         "url":"https://www.linkedin.com/...",
         "name":"Reply"
    }
}
</script>
```

Listing 5.3: Example of a *schema:ViewAction* used in emails by Linkedin using JSON-LD.

Therefore, in case an email is consumed by using the Gmail client, the various semantically annotated actions will be parsed and understood by the client, which will provide the appropriate call to action button that reflects the described action within the email markup. Of course, those actions should be annotated using the schema.org vocabulary action terms. Figure 5-4 depicts how the action button appears on the Gmail client user interface when an email includes a Gmail action in its HTML markup. The action can be implemented as shown in the example from GitHub of Listing 5.2 by using the Microdata format within the HTML or by using JSON-LD as shown in Listing 5.3, which refers to an email from the LinkedIn website that includes a view action.

5.5 Describing the generated vocabulary

The vocabulary terms discovery and the vocabulary generation as presented in sections 5.2 and 5.4, respectively, lead to the formulation of a set of terms, which is considered to be a *result vocabulary* according to the previously provided Definition 13. The presentation of the result vocabulary is considered crucial in the scope of the presented research work as it plays the role of educational material to the user of the methodology specific to the given webpage, as it facilitates the deeper understanding of the vocabulary space and the semantic annotations paradigm. Therefore, a transparent representation of the result vocabulary enables the user to understand better the process of discovering vocabulary terms by studying the connections between the used keywords and the selected vocabulary terms.

In this respect, the generated set of terms from the presented approach is a new vocabulary that is provided to the user with all the needed metadata about the terms' ranking and mappings to the webpage content in the form of keywords. To support this form of output, a new vocabulary has been designed, that could be described as mostly a technical vocabulary that facilitates the output of the discovery results to the user.

The vocabulary itself combines existing vocabularies and introduces a new names-

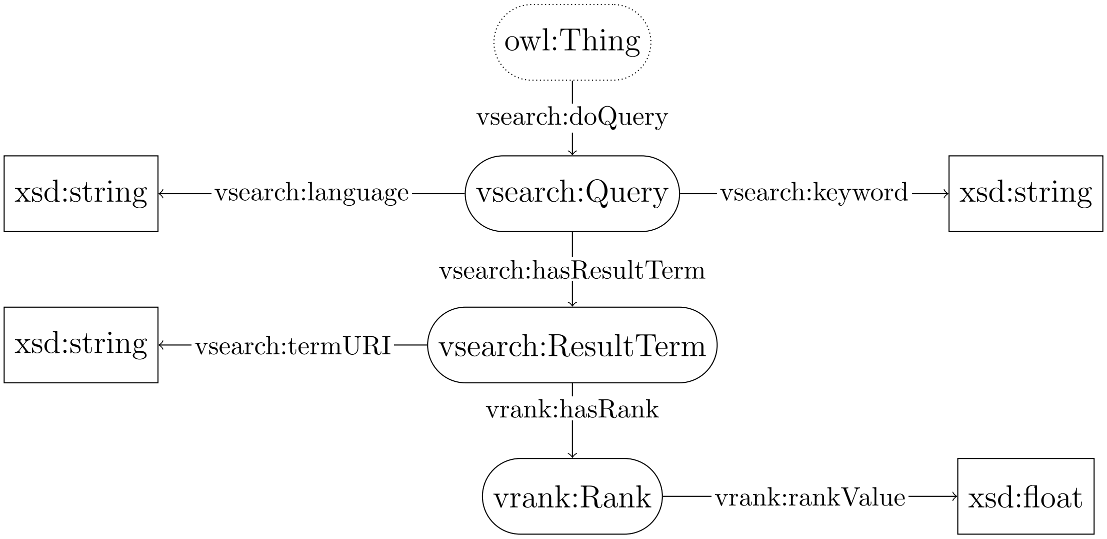


Figure 5-5: The vSearch vocabulary (published in [80]).

pace (i.e. vsearch:) for the properties and classes that weave them together with the existing vocabulary properties and classes. The relationships among the vSearch properties and classes is depicted in Figure 5-5. The purpose of the vSearch vocabulary is to provide the appropriate properties for the description of a search query with the related results and the accompanied ranking. For the ranking properties, the vRank vocabulary[9] is being reused, which is described in [66]. A structured presentation and the corresponding source in Turtle [15] or RDF/XML of the vSearch vocabulary is available under the persistent URL http://purl.org/vsearch, which redirects to http://vocab.sti2.at/vsearch.

At the core of the vocabulary, the main entity, the *vsearh:Query*, could have 1 or more keywords (using the *vsearch:hasResultTerm* object property) and 1 or more results (using the *vsearch:hasRank* object property), which are instances of the *vsearch:ResultTerm* type. Each result is connected with a *vrank:Rank* instance from the *vrank* vocabulary accompanied with the *vrank:rankValue* property. Additionally the *vsearch:ResultTerm* is mapped with a 1:1 relationship with a URI that identifies the term, via the *vsearch:termURI* datatype property. Finally, equally important is the object property *vsearch:doQuery* which enables any object to be connected with a query entity for the description of a query execution and the accompanied results.

[9] http://lov.okfn.org/dataset/lov/vocabs/vrank

The list below describes all the terms of the vocabulary and their purpose in the schema.

Thing

The *owl:Thing* class is the higher class at the hierarchy of the vocabulary classes according to the OWL W3C recommendations[10]: *The class with identifier owl:Thing is the class of all individuals.* Based on this assumption the query objects can be connected with any existing vocabulary class instance through the property that is described right afterwards, the *vsearch:doQuery* object property.

doQuery

Object property that is used to allow any entity to be connected with a query instance *vsearch:Query* and describe an executed query together with the results.

Query

Depicts the query executed against the Linked Open Vocabulary space.

hasResultTerm

Object property to map the *vsearch:Query* object to the *vsearch:ResultTerm* object. A *vsearch:Query* object could have more than one *vsearch:ResultTerm* mapped to it through the *vsearch:hasResultTerm* property. For example, the keyword “place”, as shown in Table 5.1, has three most relevant results, i.e. schema:Place, event:Place and dbpedia-owl:Place. In this case, we would expect to see three instances of *vsearch:ResultTerm* connected to a single *vsearch:Query* instance.

ResultTerm

The main class of the vsearch vocabulary that describes the vocabulary term from the LOV space that matches to the given query keyword.

[10] https://www.w3.org/TR/2004/REC-owl-semantics-20040210/syntax.html#owl_Thing_syntax

keyword

Datatype property which is used to describe the keyword of the query.

language

Datatype property defining the language of the keyword. The result term refers to the "normalised" English version of the given keyword. Throughout the examples we simplify the language concerns by assuming that both input and output are in English "en". In case the *vsearch:keyword* is given in another language than English, the output would include the English translation that is used in the discovery for consistency as another *vsearch:keyword* instance of the same query.

termURI

Datatype property with range the XML Schema String type that defines the URI of the term that matches to the given *vsearch:keyword* of the *vsearch:Query*.

The definition of the *vsearch* schema of Figure 5-5, reuses the *vRank* vocabulary namespace to describe the ranking of the result term. Every *vsearch:ResultTerm* has an object property *vsearch:hasRank*, which connects the proposed vocabulary term with the ranking score it has been assigned by the recommendation process and described by an instance of *vrank:Rank*. The ranking score range is the set of real numbers, represented by *xsd:float*[11] in the vocabulary. It is not expected to see negative values in the ranking object, but for the sake of making the schema more general, it is not limited to the non-negative floats' subset. The *vRank* vocabulary contains many more terms than those that are reused and are intended to be used in scenarios where ranking processes take place. It includes many more terms, than the adopted terms in *vSearch*, mostly about information regarding the ranking algorithm, which is not adding value in our case. Additionally, the best practices of vocabulary reusability do not dictate the complete incorporation of an existing vocabulary in a

[11] XML Schema float: https://www.w3.org/TR/xmlschema-2/#float

new one that is synthesised with composition on top of it.

The example of Listing 5.4 demonstrates the description of a vocabulary term search and the corresponding result. The example is following the Turtle[12] syntax, which makes RDF presentation easier.

```
@prefix s: <http://vocab.sti2.at/vsearch> .
@prefix v: <http://vocab.sti2.at/vrank> .

[a s:query ;
   s:language "en" ;
   s:keyword "ingredient" ;
   s:hasResultTerm [
         s:termURI "http://schema.org/ingredients" ;
         v:hasRank [ v:rankValue "5"^^xsd:float]
         ]
] .
```

Listing 5.4: Example of a *vSearch:Query* instance for a vocabulary term search in Turtle notation.

In addition to the example of Listing 5.4, the example of Listing 5.5 represents a general search described using the *vSearch* vocabulary. Although the vocabulary is able to be used to describe any type of search, throughout the manuscript it is only employed for the representation of the vocabulary term searches and the *result vocabulary* of the methodology.

```
@prefix s: <http://vocab.sti2.at/vsearch> .

[a s:query ;
   s:language "en" ;
   s:keyword "turtle recommendation" ;
```

[12] Turtle W3C recommendation: https://www.w3.org/TR/turtle/

```
    s:hasResultTerm [ s:termURI "https://www.w3.org/TR/
        turtle/" ]
] .
```

Listing 5.5: Example of a *vSearch:Query* instance for a non-vocabulary search in Turtle.

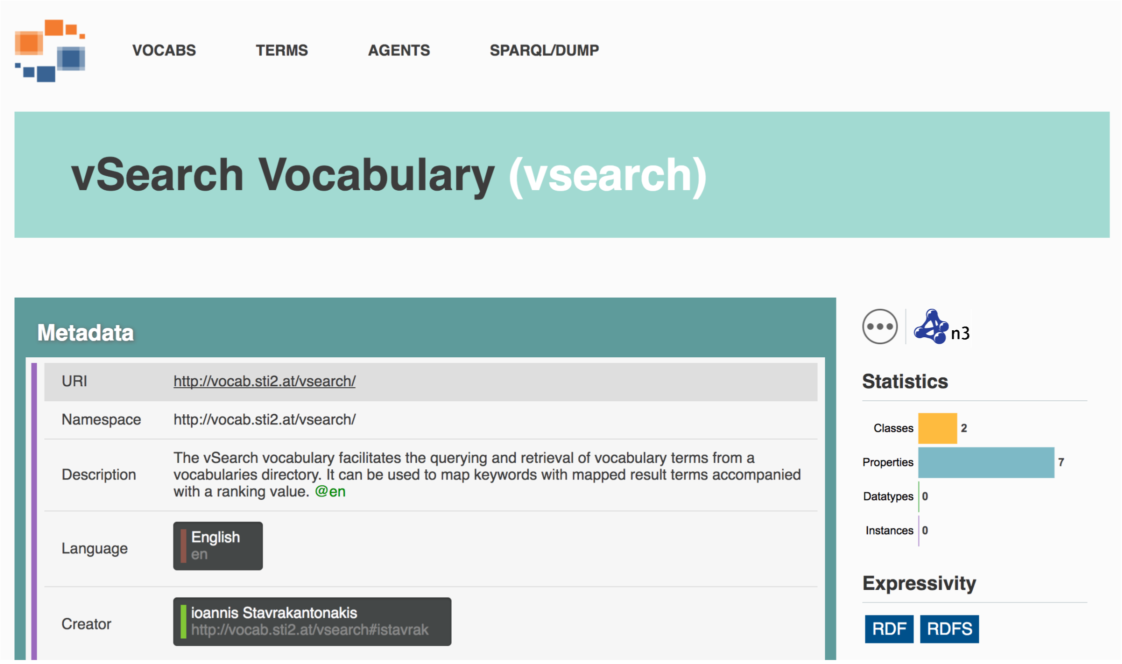
The aforementioned examples of the new vocabulary introduced under the name *vSearch*, aim to demonstrate the usage of it to represent results of a performed search, either a vocabulary search on a vocabulary directory or any other type of search that yields results. Additionally, the same vocabulary can be used to describe a search that a user would like to perform. In summary, the presented schema is a way to structurally present a search query. Therefore, within the proposed approach is not only used to provide the description of the *result vocabulary* T, but also to express the various communication messages, i.e. requests and responses, between a client and an instance of the methodology implementation as it is later presented in the implementation, Chapter 6, under Section 6.2.

The *vSearch* is listed in the LOV repository[13] as shown in Figure 5-6.

5.6 Summary

Recalling the four main requirements that a web agent should fulfil according to Hendler [39]: *communicative, capable, autonomous* and *adaptive*, as it was discussed before, we realise that the semantic annotations together with the annotation of potential actions of the defined entities promote the heart of the Semantic Web vision, the web agents. Putting all the above together and producing content that is machine understandable, it opens up a tremendous potential, which is a matter of time to be materialised within the implementation of a autonomous web agent. In this direction, Facebook has announced in 2015 [53] that the research team is working

[13] The vSearch vocabulary on LOV: `http://lov.okfn.org/dataset/lov/vocabs/vsearch`



vSearch Vocabulary (vsearch)

Metadata

URI	http://vocab.sti2.at/vsearch/
Namespace	http://vocab.sti2.at/vsearch/
Description	The vSearch vocabulary facilitates the querying and retrieval of vocabulary terms from a vocabularies directory. It can be used to map keywords with mapped result terms accompanied with a ranking value. @en
Language	English en
Creator	ioannis Stavrakantonakis http://vocab.sti2.at/vsearch#istavrak

Figure 5-6: The LOV profile of the *vSearch* vocabulary.

and testing a digital assistant that will live within the messenger application that the users use to chat with their friends. Thus, the users will have the ability to talk to the digital assistant and ask various questions in the same way that they talk to their friends. The digital assistant is named M and as it is mentioned in the Facebook post [53]: *"Unlike other AI-based services in the market, M can actually complete tasks on your behalf. It can purchase items, get gifts delivered to your loved ones, book restaurants, travel arrangements, appointments and way more"*. Therefore, M will be a communicative assistant, as it has the ability to participate in a discussion with the user; it will be capable to complete tasks, like reservations, purchases, etc.; and it will be autonomous, as it will not need to be assisted by a human on every interaction. These are the insights that a reader can extract from the given post, which lead to the conclusion that *M will be one of the first widely deployed Web agents* that will be tested and support millions of people, based on the figures about the active users of Facebook. Other approaches already exist, like the Google Now, Siri from Apple, and Microsoft Cortana. Although Siri is able to make restaurant reservations in the US by using the OpenTable application, all of them are mostly used to retrieve information, review the user's calendar, etc. They do not focus on performing actions in contrast to the planned functionality of M as long as it has been advertised at the time this manuscript is authored.

The aforementioned capabilities of the Facebook digital assistant M or the Amazon Echo, would not be possible without machine understandable content on the Web resources that the system will have integrated and based on. Probably, the various digital assistants will be using API calls to execute various actions like reservations (e.g. by using OpenTable for the user's dinner arrangement), but those actions could be even defined within the restaurants webpages making it accessible for agents like M. The vision of Web agents completing tasks in the near future is already happening. In this direction, the presented framework aims to fulfil a pragmatic need of helping the semantic annotation development process, by recommending vocabulary terms appropriate for the presented content based on the LOTR ranking, by recommending vocabulary terms for the static parts as that was previously defined (i.e. media types) and by discovering actions that could performed on top of the webpage entities.

Chapter 6

Approach implementation

Development reference and usage guide

The implementation of the described methodology is completely decoupled from the approach. Thus, the theoretical part and the various algorithms are not affected by any limitations and optimisations of the developed framework. In the scope of the research work, the designed approach has been implemented as a Web Service. Furthermore, one of the basic ideas behind the architecture of the framework is the modularity of the various components. In this respect, the implementation allows the replacement of any of the modules with an external module that the prospective host of the framework would like to provide to the endpoint users. The only prerequisite is to develop the correspondent adaptor to the interface of the respective module. As it has already been clarified by the aforementioned descriptions, this chapter outlines the technical parts and the architecture of the implementations that accompanies the research work.

Therefore, the sections of this chapter focus on the description of the implementation of the research methodology behind the LOVR framework. The implemented Web Service has been named *vocab-recommender*, with only reason behind the difference in the naming to be the clarity that the *vocab-recommender* could support not only the LOVR framework's algorithms but any respective modules that a developer would like to use in order to substitute the LOVR framework. This decision follows the *open architecture*[1] principles, which specify that it should be easy to add, update and swap components in an architecture in order to be considered as open. The aim of *vocab-recommender* is to provide vocabulary terms recommendations by employing a recommendation methodology. In the published version of the *vocab-recommender*, the LOVR framework is at the core of the implementation, but more modules are integrated in order to provide the final result. For example, the description of the results is based on the *vSearch* vocabulary, which was presented in Section 5.5, and is one of the contributions of the presented research work. Therefore, a software engineer could extend the *vocab-recommender* tool to use a different recommendation framework, while keeping the presentation of the results following the *vSearch* vocabulary and structure.

The chapter begins with the presentation of the architecture and the various modules that consist the *vocab-recommender* framework. Also, the communication of the modules and the data flow is explained within the first section, Section 6.1. The following section, Section 6.2, plays the role of the usage reference of *vocab-recommender*, as it describes the various endpoints provided by the Web service. Finally, Section 6.3 presents the technology stack that has been used to implement *vocab-recommender*, while Section 6.4 summarises the implementation details.

6.1 Architectural design

The implementation of the methodology follows the Web service paradigm, which makes *vocab-recommender* available to be used either by humans or by any solution that has integrated it. Figure 6-1 shows the simplicity of the usage, which is based on a request-response communication between the user and the framework. The various components described in the diagram of Figure 5-1 reflect the modules that need to be implemented as part of the provided Web service in order to produce recommendations of vocabulary terms for a given website.

In this regard, exploring deeper the architecture behind the Web service, the

[1] Open architecture: https://en.wikipedia.org/wiki/Open_architecture

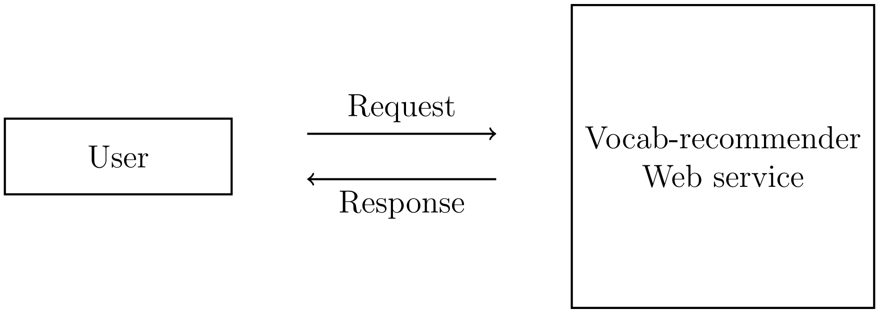


Figure 6-1: Vocab-recommender provided as a Web service.

architectural diagram of Figure 6-2 showcases the various modules and layers that comprise the LOVR framework and the *vocab-recommender* framework. Three main layers are distinguished within the designed framework, i.e. a) the Web service layer that is responsible for the interaction with the external agents that consume the service, responsible with the communication with the backend system that will handle any incoming requests and the propagation of the *result vocabulary* $\mathcal{T}$ to the end user; b) the keywords extraction layer, which is responsible for the generation of the keyword list that is used at the vocabulary terms discovery; and c) the LOVR framework layer that include all the components that search and rank the terms and vocabularies in order to produce the final list of the terms that comprise the *result vocabulary*. The second layer that is related to the extraction of keywords is not mandatory and it is skipped in case the request is explicitly defining the list of keywords.

The LOVR layer consists of four main components that are being orchestrated by the framework as shown later in the sequence diagram of Figure 6-3 to discover vocabulary terms for a list of keywords. The *Searcher* is responsible for the search of vocabularies and vocabulary terms by leveraging any external sources that have been integrated. The module can connect to a local RDF database that includes raw files about vocabularies from the LOV repository[2] or any other vocabulary provider. Another option, is to utilise the LOV API endpoint[3] via the *Searcher*.

At the core of the LOVR framework, the *Ranker* module is responsible for real-

[2] LOV data: http://lov.okfn.org/dataset/lov/sparql

[3] LOV API: http://lov.okfn.org/dataset/lov/api

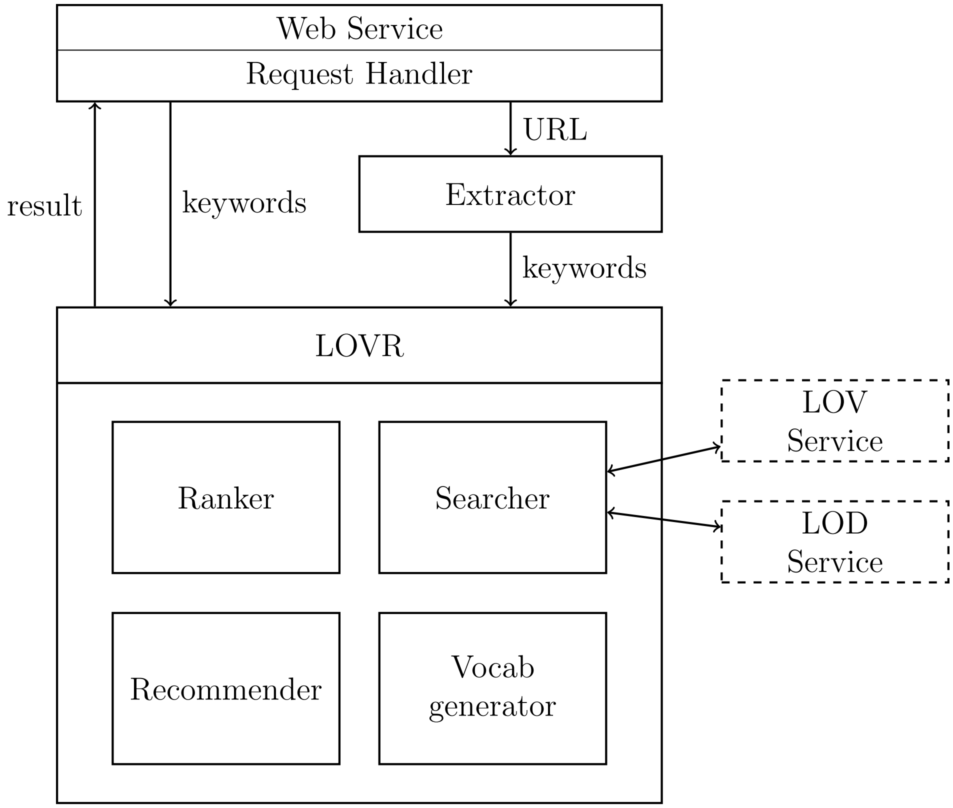


Figure 6-2: The developed *vocab-recommender* framework architecture depicting the various modules grouped in the various layers of the solution. The Web service can either handle a webpage URL or a list of keywords. The dashed elements refer to external services that have been integrated.

ising all the theoretical parts that have been designed in the scope of the proposed methodology, as presented in the previous two chapters, i.e Chapter 4 and Chapter 5. *Ranker* consists of more than one ranking methods and metrics. It is responsible for ranking both the vocabularies and the vocabulary terms. The former is conducted in an asynchronous way and not at every search, based on the assumption that new vocabularies are added to the vocabulary space in a significant low rate.

The *Recommender* module is the one that orchestrates the invocation of the searches and the ranking of the terms in order to respond with a sorted list of result terms to the upper layers that propagate up to the *vocab-recommender* Web service response returned by the *Request Handler*. The *result vocabulary* is generated by the *Vocab generator* depicted in Figure 6-2, which is responsible to generate instances of the *vSearch* classes to describe the result terms.

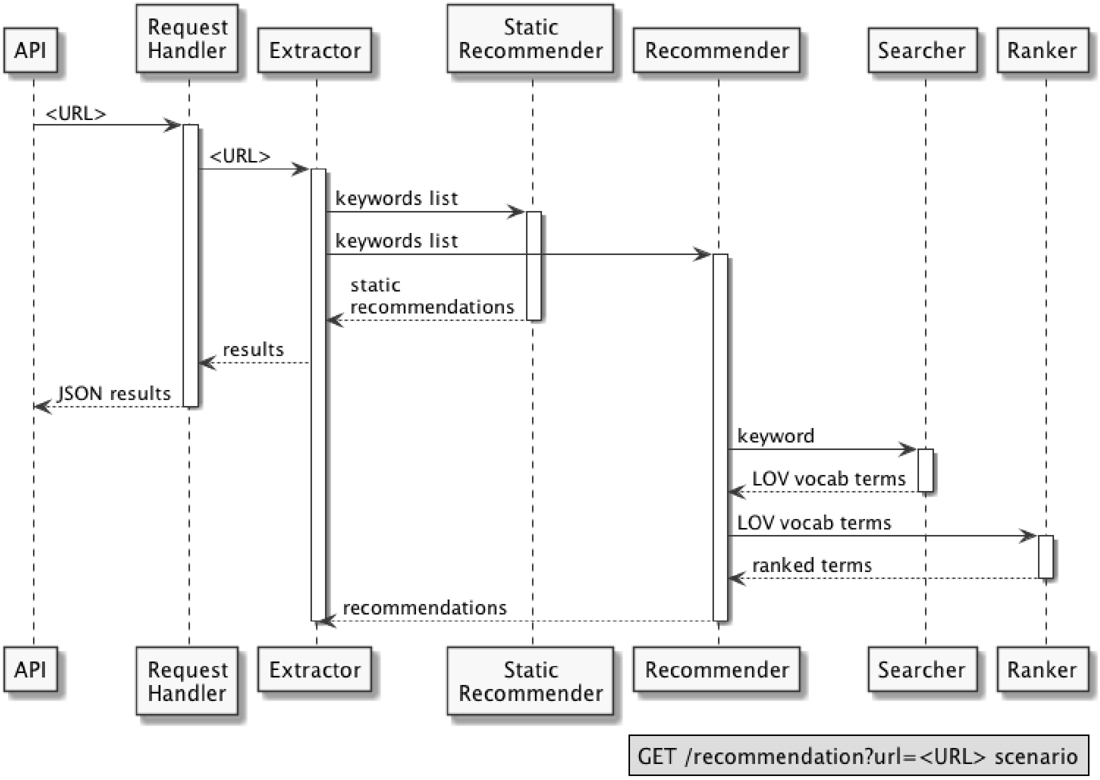


Figure 6-3: The depicted sequence diagram demonstrates the various modules that are orchestrated to produce the recommendation output based on a given URL. In a different scenario that we provide a list of keywords, the extractor is skipped. As published in [81].

Listing 6.2 showcases the response of a search against the *vocab-recommender* endpoint that accepts a list of keywords. As show in the sample, the response is based on the JSON format.

6.2 Framework usage

The architectural design was described within the previous section by explaining the various components and their responsibilities. Figure 6-1 shows the usage scenario of the Web service and the amount of interactions that are needed for a user in order to complete a successful session. In brief, all it is required is the formation of a request to the Web service. The Web service can be invoked either using a *GET* request or via a *POST* request. In the case of the former, the request is similar to the one shown in Listing 6.1, which includes the input parameters at the query string of the URL. A comprehensive list of the Web service endpoints is presented as part of Section 6.2.1.

```
GET http://ist-lab.sti2.at/vocab-recommender/
   recommendation?url=http://istavrak.com
```

Listing 6.1: Example of an endpoint from the framework deployed on the lab server.

Furthermore, moving towards the employment of the *vSearch* vocabulary in the scope of the methodology, Listing 6.2 showcases the usage of the *vSearch* vocabulary in JSON-LD format. This format is the suggested format to be used throughout the communication with the *vocab-recommender* API. An example of a request is shown in Listing 6.3, which requests results for two keywords.

The JSON-LD format is the main way that is promoted throughout the presented examples for the communication between the server and the client. Main reason is the context definition that helps to self-describe the schema that the data follows within the JSON structures. More information about the JSON-LD format have already been described in Section 2.2.1. The remainder of the section is divided into two parts, i.e. the description of the *vocab-recommender* endpoints (API) and the

```
{
  "@context": {
    "vrank": "http://vocab.sti2.at/vrank#",
    "vsearch": "http://vocab.sti2.at/vsearch#",
    "xsd": "http://www.w3.org/2001/XMLSchema#",
    "vrank:rankValue": {
      "@type": "xsd:float"
    }
  },
  "vsearch:query": {
  "vsearch:keyword": "ingredient",
  "vsearch:language": "en",
  "vsearch:hasResultTerm": {
      "vsearch:termURI": "http://schema.org/ingredients",
      "vrank:hasRank": {
        "vrank:rankvalue": 5
      }
  }}
}
```

Listing 6.2: Example of a vSearch:Query instance for a response in JSON-LD.

```
{
  "@context": {
    "vsearch": "http://vocab.sti2.at/vsearch#"
  },
  "vsearch:doQuery": [
    {"vsearch:query": {
    "vsearch:keyword": "ingredient",
    "vsearch:language": "en"}
    },
    {"vsearch:query": {
    "vsearch:keyword": "serving",
    "vsearch:language": "en"}
    }
  ]
}
```

Listing 6.3: Example of a vSearch:Query instance for a request in JSON-LD. Thus, we see only information about the query keywords and not the respective result terms and ranking.

```
GET /recommendation?url=<url>[&static={true|false}]
```

Listing 6.4: The URL-based recommendation HTTP GET request endpoint of the *vocab-recommender* Web Service. Table 6.1 specifies the parameters.

```
/* Request */
GET /recommendation?url=www.example.com&static=false

/* Response */
{"doQuery":
  [{

  }],"success":true}
```

Listing 6.5: The basic recommendation HTTP GET request endpoint of the *vocab-recommender* Web Service. The input parameters include the URL of the target webpage and the static flag set to false in order to exclude them from the final result. Thus, the response only includes the core tokens that appear in the document.

presentation of a possible UI that could consume the Web service, respectively in Section 6.2.1 and Section 6.2.2.

6.2.1 Vocab-recommender API

The public endpoints of the framework are exposed via a RESTful Web Service[4] tier. The REST paradigm is based on a fundamental asset of the Web, the HTTP protocol by leveraging the methods of it, i.e. GET, POST, PUT, etc. Therefore, to retrieve a resource, a Web Service that is following the REST paradigm should support an HTTP GET request like the one in Listing 6.4 and to create a resource on the server, it should support an HTTP POST request like the one in Listing 6.6.

Getting recommendations for a target URL

This is the basic and main request that the Web Service has been developed for. It accepts a URL as parameter, which is the *target URL* that the framework will produce recommendations for. The URL format for this request is shown in Listing 6.4.

[4] http://www.ibm.com/developerworks/library/ws-restful/

Listing 6.5 demonstrates the response to the most basic endpoint request, which accepts as input parameter a webpage URL. The response includes all the extracted tokens that have been used as keywords to generate the set of vocabulary terms. The retrieved vocabulary terms are accompanied by the ranking score, which is an important indicator to the end user about the matching of the suggested term to the target keyword. Furthermore, the output set of terms includes a subset of the various static elements that can be detected by the extractor module. Additionally, there is a flag parameter which allows to disable the inclusion of vocabulary terms about the static elements, i.e. "static", and it is set to true by default. In the next listing, there is an example of a request that provides the keywords to search for terms and it cannot be combined with the "static" parameter as in this case the framework skips the tokens extraction phase.

Getting recommendations for a list of keywords

Besides the aforementioned endpoint, the *vocab-recommender* Web Service offers one more endpoint, which allows to request recommendations only for keywords that the user decided as relevant. As shown in Listing 6.6, the request accepts as query string parameters a list of keywords separated by comma. In this way, the first layer of the framework, which is responsible for extracting keywords and special HTML elements is bypassed. The output response is identical to the one triggered by the URL-based request. Another possibility would be to construct a *POST* requeset at the same URL but without any URL paramers, but within the HTTP message body including a JSON-LD snippet like the one presented in Listing 6.3. Comparing the two request methods depicted in Listing 6.6, we realise that the *POST* method is more verbose and transfers more information to the server side. For example, the language of the keywords in the *GET* request is assumed to be English, while in the *POST* method, the language is explicitly defined.

Listing 6.7 presents the request and response example, which uses the keyword based endpoint. In this example, the keywords are two multimedia types, which can be used to test the mappings that are presented in Table 5.5. Thus, the response

```
GET /recommendation?query=<keyword1,keyword2>

POST /recommendation
Body:
{ "@context": {
    "vsearch": "http://vocab.sti2.at/vsearch#" },
    "vsearch:doQuery": [
      {"vsearch:query":{"vsearch:keyword":<keyword1>,
                        "vsearch:language":<language1>}},
      {"vsearch:query":{"vsearch:keyword":<keyword2>,
                        "vsearch:language":<language2>}}
    ]
}
```

Listing 6.6: The keywords based recommendation HTTP GET and POST request endpoints of the *vocab-recommender* Web Service. Table 6.1 specifies the parameters.

includes only results about the two passed terms accompanied with the highest rank value, i.e. 1.0, as those keyword-term pairs are considered exact matched by being explicitly and manually specified.

Getting recommendations for multimedia parts

A drawback of using the *query* endpoint instead of the *url* one, is the fact that the generated vocabulary terms list will not contain any terms that could be a possible match based on the various HTML elements (called static throughout the approach) of the target webpage. For this reason, one more endpoint has been created that allows to request all the mappings of static elements to vocabulary terms. Thus, the response of the GET request shown in Listing 6.8 will return a list of vocabulary terms for media types, e.g. the image, any needed copyright statement, dimensions or EXIF data.

Posting feedback about the recommendation

Using the previously specified endpoints, the user of the Web Service will receive a response with vocabulary terms recommendations for the given input. A possible extension of the framework could include a feedback look. In order to support a feedback

```
/* Request */
GET /recommendation?query=image,video

/* Response */
{"doQuery":
  [{
    "keyword":"image",
    "hasResultTerm":{
        "termURI":"http://schema.org/image",
        "hasRank":{"rankValue":1.0}
      }
  },
  {
    "keyword":"video",
    "hasResultTerm":{
        "termURI":"https://schema.org/video",
        "hasRank":{"rankValue":1.0}
      }
  }],"success":true}
```

Listing 6.7: The keywords based recommendation HTTP GET request endpoint of the *vocab-recommender* Web Service. The response to this request will always be the same and its agnostic to the target webpage. The shown ranking score 1.0, is the maximum value of the ranking range.

```
GET /recommendation/static
```

Listing 6.8: The static recommendation HTTP GET request endpoint of the *vocab-recommender* Web Service. It refers to common webpage parts, including HTML elements that are domain agnostic.

Parameter	**Type**	**Description**
url	string	The target URL to generate suggestions for.
query	strings list	A comma separated list of keywords that represent the target content.
static	boolean	A flag to disable the inclusion of static terms in the final result.

Table 6.1: Vocab-recommender Web Service parameters description.

look the framework should includ in every response together with the recommendation and identifier that could be used as a reference identifier to the recommendation task that took place and generated the new vocabulary. In this way, the user would be able to reply back with feedback in a structured way by using an HTTP POST request, which would mention the identifier.

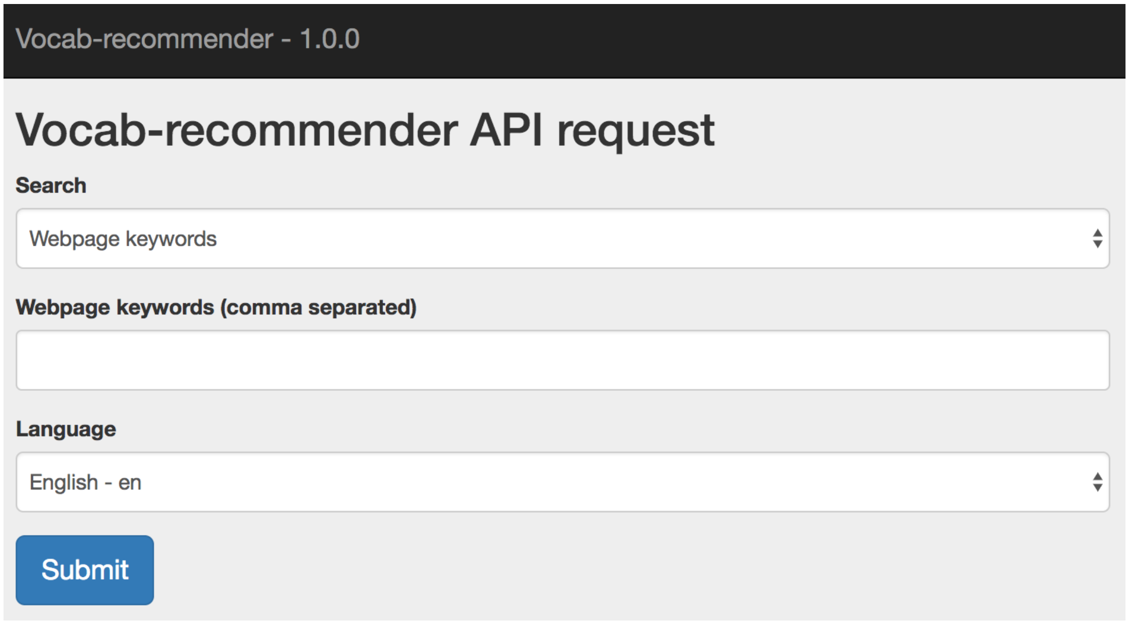
6.2.2 Vocab-recommender Web UI

All the above demonstrated endpoints can be used either by humans or Web agents in order to discover vocabulary terms. Users that are keen on building HTTP requests, executing them and parsing the responses either in a manual fashion or through some programmatic based approach, the set of endpoints suffices in order to explore the functionality of the vocabulary terms discovery assistant that *vocab-recommender* provides. Thus, the above described endpoints are considered as the API of the *vocab-recommender* Web service.

Beyond the *vocab-recommender* API, a User Interface (UI) based tool has been developed that can be used by a user to explore the functionality of the *vocab-recommender* framework. Figure 6-4 demonstrates the generation of a keyword based search request against the *vocab-recommender* Web service by using the provided User Interface. This UI form after clicking on the search button, generates a *POST* request to the implemented Web service with the given parameters. The result JSON-LD is returned to the user.

An alternative to using the presented UI would be any HTTP request client. There are plugins for Web browsers (like Chrome and FireFox), which help to send an HTTP *POST* request and command line tools that accomplish the same. For example in the UNIX systems space, curl[5] is a very popular tool for making HTTP requests. Curl is also available for the Microsoft Windows Operating System.

[5] Curl: `https://curl.haxx.se/`



Vocab-recommender - 1.0.0

Vocab-recommender API request

Search

Webpage keywords

Webpage keywords (comma separated)

Language

English - en

Figure 6-4: The depicted User Interface (UI) refers to a keyword based search, allowing the user to define the keywords in a coma separated array of values and to explicitly indicate the language of the keywords.

6.3 Technology stack

The technology stack that supports the developed endpoint is based on the Java programming ecosystem and compatible frameworks. Specifically, the development has been completed using the Java EE 7. Java 8 features are not used throughout the code for easier compatibility with older application servers.

The code repository is publicly shared under the MIT License through a Git[6] based versioning platform, the GitHub. Visiting the GitHub page of the project[7], the user is given all the information needed in order to build and deploy[8] the *vocab-recommender* Web service on an any Web server that supports the Java EE specifications.

The RESTful Web Service has been implemented using the Spring framework[9], while the User Interface (UI) tool has been implemented using Java Server Pages (JSP) together with the Spring framework. Moreover, it is worth mentioning that the

[6] Git version control system: `https://git-scm.com/`

[7] Vocab-recommender GitHub: `https://github.com/istavrak/vocab-recommender`

[8] The latest version of vocab-recommender is deployed under the URL: `http://ist-lab.sti2.at/vocab-recommender`.

[9] https://spring.io/guides/gs/rest-service/

communication with the Web Service (requests/responses) is based on the JSON and the JSON-LD formats. Examples of requests and responses are given in the listings 6.7, 6.6. The presentation of the results in the response is realised by materialising the vSearch vocabulary[10] that has been designed in the scope of this dissertation and described in Section 5.5.

The presented research mainly focuses in the discovery of vocabulary terms and not in the extraction of keywords from the input webpage. Therefore, there is no contribution in the field of Natural Languate Processing (NLP), although the NLP approaches are used in order to provide a complete implementation and framework. In the scope of the implementation, the Stanford CoreNLP [52] tool has been integrated to the vocab-recommender codebase in order to provide easy extraction of keywords from the given webpage. The NLP module is responsible for the extraction of all the unique nouns that appear in the content of the target webpage. Those extracted tokens are later used as keywords to discover vocabulary terms.

The implementation has followed a modular way in order to enable the easy development of extensions for the framework, but also to enable the connection of external services that could substitute any of the secondary modules but still important module, e.g. the Natural Language Processing (NLP) module, the various keyword extractors, etc. For example, the LOV search module can be a client that consumes the public API of LOV, or it can be querying an internal persistence layer or database which has been populated with all the vocabularies and metadata of the LOV repository.

6.4 Summary

Following the open architecture paradigm, the presented implementation aims to showcase the realisation of the research results and algorithms behind the *LOVR framework*. Thus, it described the created Web service and the architecture of it, the communication between the various components and the data flow from the input

[10] `http://vocab.sti2.at/vsearch`

until the output result vocabulary terms set, as shown in the sequence diagram of Figure 6-3 and the components diagram of Figure 6-2. Furthermore, the current chapter played the role of a user manual for the *vocab-recommender* Web service by explaining in detail how to consume the various endpoints of the Web service and what the different parameters control at the server side, like the flag for the inclusion of the multimedia HTML elements recommendations. The simple UI presented in Section 6.2.2 helps to consume the various endpoints by assisting in the formation of the HTTP requests.

It is worth mentioning, that the *vocab-recommender* endpoints can be consumed by Web applications that need a limited list of recommended terms for a complete webpage without the need to manually search each one of the keywords in the LOV directory and decide among the search results for the best matches. Therefore, *vocab-recommender* can also play the role of an aggregator of all the searches needed to extract vocabulary terms from LOV for a given webpage.

Chapter 7

Use Cases

Applying the approach to real world scenarios

This chapter aims to examine the application of the approach against real world scenarios by analysing what parts of the webpages could be annotated and what benefits the semantic annotations would bring. In addition, the approach is used against the use cases in order to evaluate how the approach meets the expectations and hypotheses that were defined at the first place during the description of each one of them. However, the goal is not to extensively evaluate the approach, but rather to see how it works in the presented use cases by using specific example webpages from the Web. A more detailed evaluation of the approach is the topic of the next chapter, Chapter 8.

Therefore, the four following use cases demonstrate the effectiveness of the approach in real world scenarios, which is the aim of this chapter. The use cases have been chosen to cover various major domains, i.e. a) the local businesses domain, which is presented in Section 7.1; b) the food recipe domain, presented in Section 7.2; c) the cultural heritage domain, as presented in Section 7.3 and d) the online article publication domain, represented by Section 7.4. Last but not least, the use case of a recipe page is consider an important part of the conducted experiments as recipes was one of the first types of content that search engine providers used to support semantic search for. In addition, the four discussed use cases map to the four domains

Label number	**Information**
1(a)	Title of the room
1(b)	Room type
2	Description
3	Amenitites
4	Pricing
5	Seasons pricing
6	Image

Table 7.1: Analysis of the various information bits on the hotel room webpage that can be annotated using the appropriate vocabulary terms.

that were distributed within the survey for manually discovering vocabulary terms of Section 3.3.

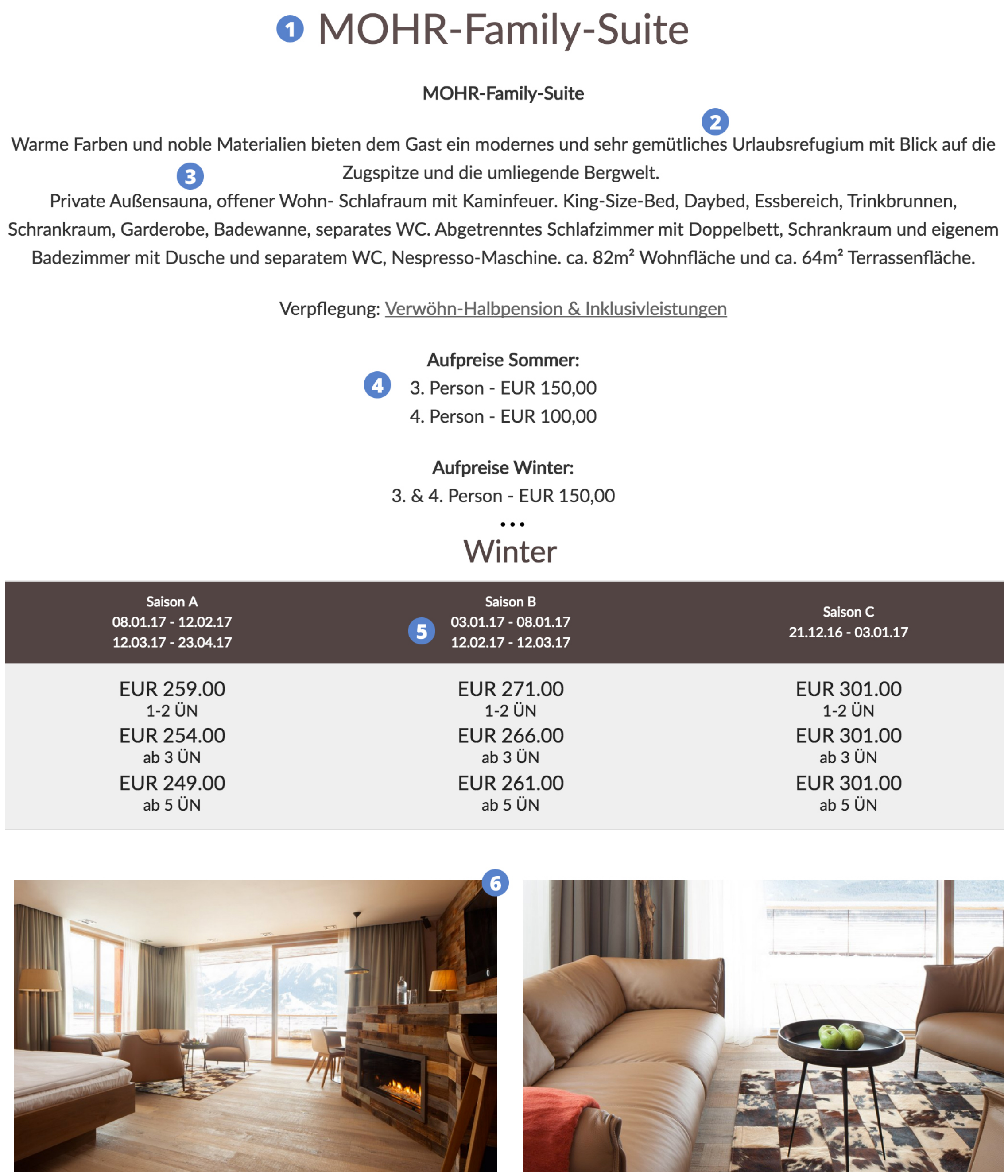
Each one of the use cases is presented in the respective section by presenting a use case representative example from the Web. The description of the use case includes a screenshot of the webpage together with some visual labels, which allow the discussion of the various parts of the content. Furthermore, having annotated the aforementioned parts of the webpage under question, allows the answering of several questions by having the information in a semantically structured way. An example set of questions is enlisted at each one of the use cases.

7.1 Local business page

The local business webpage use case refers to the Web presence of a local business entity. For the example of Figure 7-1, a randomly selected hotel webpage from Austria is being used. A hotel room webpage has some important content worth annotating in order to transform it to a structured Web entity. Table 7.1 analyses the various data points that can be annotated on the webpage, while Figure 7-1 marks the various areas that the identifiers of Table 7.1 refer to.

The following sample questions refer to possible inquiries that could be answered using the content presented on the webpage.

- What is the price per night for August?



MOHR-Family-Suite

MOHR-Family-Suite

Warme Farben und noble Materialien bieten dem Gast ein modernes und sehr gemütliches Urlaubsrefugium mit Blick auf die Zugspitze und die umliegende Bergwelt.

Private Außensauna, offener Wohn- Schlafraum mit Kaminfeuer. King-Size-Bed, Daybed, Essbereich, Trinkbrunnen, Schrankraum, Garderobe, Badewanne, separates WC. Abgetrenntes Schlafzimmer mit Doppelbett, Schrankraum und eigenem Badezimmer mit Dusche und separatem WC, Nespresso-Maschine. ca. 82m² Wohnfläche und ca. 64m² Terrassenfläche.

Verpflegung: Verwöhn-Halbpension & Inklusivleistungen

Aufpreise Sommer:
3. Person - EUR 150,00
4. Person - EUR 100,00

Aufpreise Winter:
3. & 4. Person - EUR 150,00

...

Winter

Saison A 08.01.17 - 12.02.17 12.03.17 - 23.04.17	Saison B 03.01.17 - 08.01.17 12.02.17 - 12.03.17	Saison C 21.12.16 - 03.01.17
EUR 259.00 1-2 ÜN	EUR 271.00 1-2 ÜN	EUR 301.00 1-2 ÜN
EUR 254.00 ab 3 ÜN	EUR 266.00 ab 3 ÜN	EUR 301.00 ab 3 ÜN
EUR 249.00 ab 5 ÜN	EUR 261.00 ab 5 ÜN	EUR 301.00 ab 5 ÜN

Figure 7-1: Screenshot from a hotel webpage with the annotation placeholder labels.

Keyword	Result vocabulary term	Comment	Information
Suite	schema:Suite		1(b)
Sauna	lgdo:Sauna		3
Bed	acco:bed		2
Season	schema:season	Irrelevant	-
-	dbpedia-owl:price	Pattern based	4, 5
-	schema:name	HTML H1	1(a)
-	schema:ImageObject	Multimedia term	6

Table 7.2: Result vocabulary of the hotel room webpage use case. The last column corresponds to the label number of the information that is expected to be annotated as described in Table 7.1 and in Figure 7-1.

- How do the rooms look like?

- Does it include sauna?

- Does the suite room type accommodate 5 persons?

The above questions are only a few that can be answered with the webpage content when that is annotated. Thus, an agent would incorporate the values of the vocabulary terms that it can recognise and it would do the related calculations in order to properly respond to the requests. Obviously, it needs to be aware about the way that the respective values can be used, e.g. what is the formula for the price that should be used in order to calculate the cost for a reservation of a given number of days.

In the scope of the use case presentation, the approach has been applied against the webpage of the use case in order to generate the *result vocabulary* that corresponds to it. The result is presented in Table 7.2 and not as a JSON object for readability reasons. As shown in Table 7.2, the approach was able to cover all the parts that were expected to be annotated. It is worth highlighting that the pattern knowledge base and the static enrichment (multimedia, HTML structure) have a significant impact on the final result vocabulary.

Label number	Information
1	Title of the recipe
2	Rating
3	Number of servings
4	Total time needed
5	Preparation time
6	Image
7	Ingredients
8	Execution steps
9	Nutritional info

Table 7.3: Analysis of the various information bits on the recipe webpage that can be annotated using the appropriate vocabulary terms.

7.2 Recipe page

The second use case is taken from a recipe website. The recipe domain is the one of the first domains to utilise the power of structured data. Google back in 2011, announced[1] the *Recipe View* for the facilitation of recipes search accross the Web. *Recipe View* allowed the users to narrow down the search of a recipe by indicating the ingredients that should be included or not in the recipes that appear in the result set. However, this search dimension is discontinued[2] at the time of this dissertation's writing. In this scope, the webpage depicted in Figure 7-2 is taken from the allrecipes.com[3] website and describes a recipe preparation for a pizza.

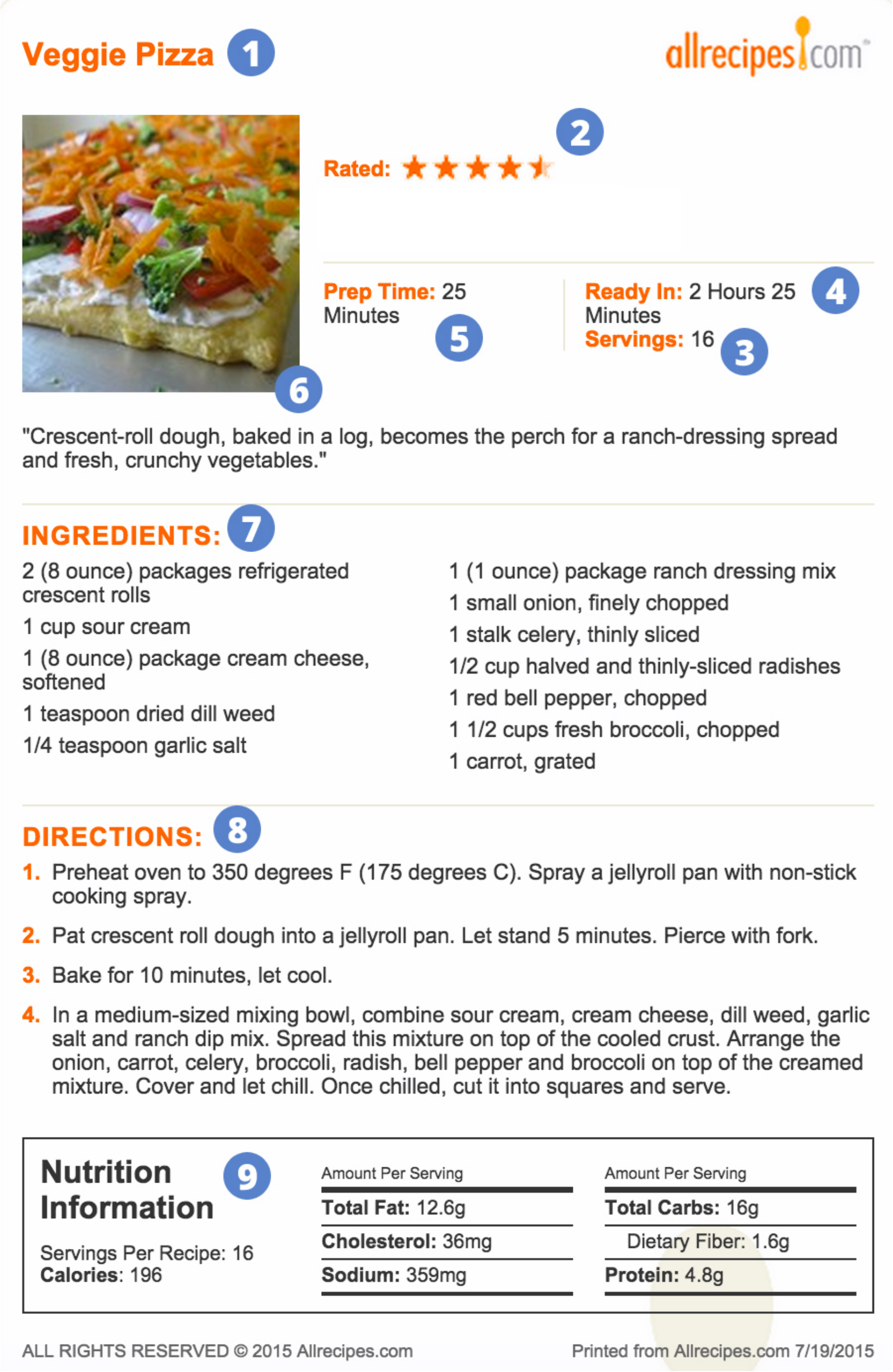
The following sample questions refer to possible inquiries that could be answered using the content presented on the webpage.

- What is the preparation time of the recipe?
- Does the recipe contain onions?
- How many servings does it provide?
- Are the calories per serving more than 200?

[1] Google official blog about Recipe View: `https://googleblog.blogspot.de/2011/02/slice-and-dice-your-recipe-search.html`

[2] Google official blog about recipe search: `http://googlesystem.blogspot.de/2014/01/google-streamlines-search-options.html`

[3] Allrecipes pizza recipe: `http://allrecipes.com/Recipe/Veggie-Pizza`



Veggie Pizza

Rated: ★★★★½

Prep Time: 25 Minutes

Ready In: 2 Hours 25 Minutes
Servings: 16

"Crescent-roll dough, baked in a log, becomes the perch for a ranch-dressing spread and fresh, crunchy vegetables."

INGREDIENTS:

2 (8 ounce) packages refrigerated crescent rolls

1 cup sour cream

1 (8 ounce) package cream cheese, softened

1 teaspoon dried dill weed

1/4 teaspoon garlic salt

1 (1 ounce) package ranch dressing mix

1 small onion, finely chopped

1 stalk celery, thinly sliced

1/2 cup halved and thinly-sliced radishes

1 red bell pepper, chopped

1 1/2 cups fresh broccoli, chopped

1 carrot, grated

DIRECTIONS:

1. Preheat oven to 350 degrees F (175 degrees C). Spray a jellyroll pan with non-stick cooking spray.
2. Pat crescent roll dough into a jellyroll pan. Let stand 5 minutes. Pierce with fork.
3. Bake for 10 minutes, let cool.
4. In a medium-sized mixing bowl, combine sour cream, cream cheese, dill weed, garlic salt and ranch dip mix. Spread this mixture on top of the cooled crust. Arrange the onion, carrot, celery, broccoli, radish, bell pepper and broccoli on top of the creamed mixture. Cover and let chill. Once chilled, cut it into squares and serve.

Nutrition Information

Servings Per Recipe: 16
Calories: 196

Amount Per Serving	Amount Per Serving
Total Fat: 12.6g	**Total Carbs:** 16g
Cholesterol: 36mg	Dietary Fiber: 1.6g
Sodium: 359mg	**Protein:** 4.8g

Printed from Allrecipes.com 7/19/2015

Figure 7-2: Screenshot from the allrecipes.com page with the annotation placeholder labels.

Keyword	Result vocabulary term	Comment	Information
Recipe	schema:Recipe		1
Rated	schema:Rating		2
Serving	schema:servingSize		3
Ingredient	schema:ingredients		7
Directions	dbpedia-owl:CardinalDirection	Irrelevant	-
Nutrition	schema:nutrition		9
-	schema:ImageObject	Multimedia term	6
-	schema:Duration	Pattern based	4,5
-	schema:name	HTML H1	1

Table 7.4: Result vocabulary of the recipe webpage use case. The last column corresponds to the label number of the information that is expected to be annotated as described in Table 7.3 and in Figure 7-2.

Providing the recipe content in a structured data format allows the search engines to answer questions from the pool of recipes in a more accurate way. A search engine could leverage the various types of information shown in Table 7.3 in order to effectively filter out recipes that do not fulfil the search criteria. For example searching for a pizza recipe that needs less than 30 minutes of preparation woudl not be possible without having that information structurely presented on the webpage.

The selected webpage includes already semantic annotations, if we check the HTML source of it. Later during the evaluation process the existing annotations can be used for comparison, in terms of comprehensiveness, with the *result vocabulary* that the proposed methodology generates.

The approach has been applied against the webpage of the use case in order to generate the *result vocabulary* that corresponds to it. The result is presented in Table 7.4 and not as a JSON object for readability reasons.

Studying the results of Table 7.4, we can realise the effectiveness of the approach, by comparing the keywords that were used and the recommendations that the use case is provided with. The webpage uses the word “Directions” to describe the recipe instruction steps section, which is very hard to map to the keyword that the table shows by only searching with this keyword in the vocabulary space. However, it would be possible to be retrieved by leveraging historical data or a knowledge base with various domain models (including the recipe case). Similarly to the hotel use

Label number	**Information**
1	Title
2	Picture
3	Location
4	Entrance fee
5	Opening hours
6	Contact info
7	Description
8	Exhibition dates

Table 7.5: Analysis of the various information bits on the Louvre exhibition webpage that can be annotated using the appropriate vocabulary terms.

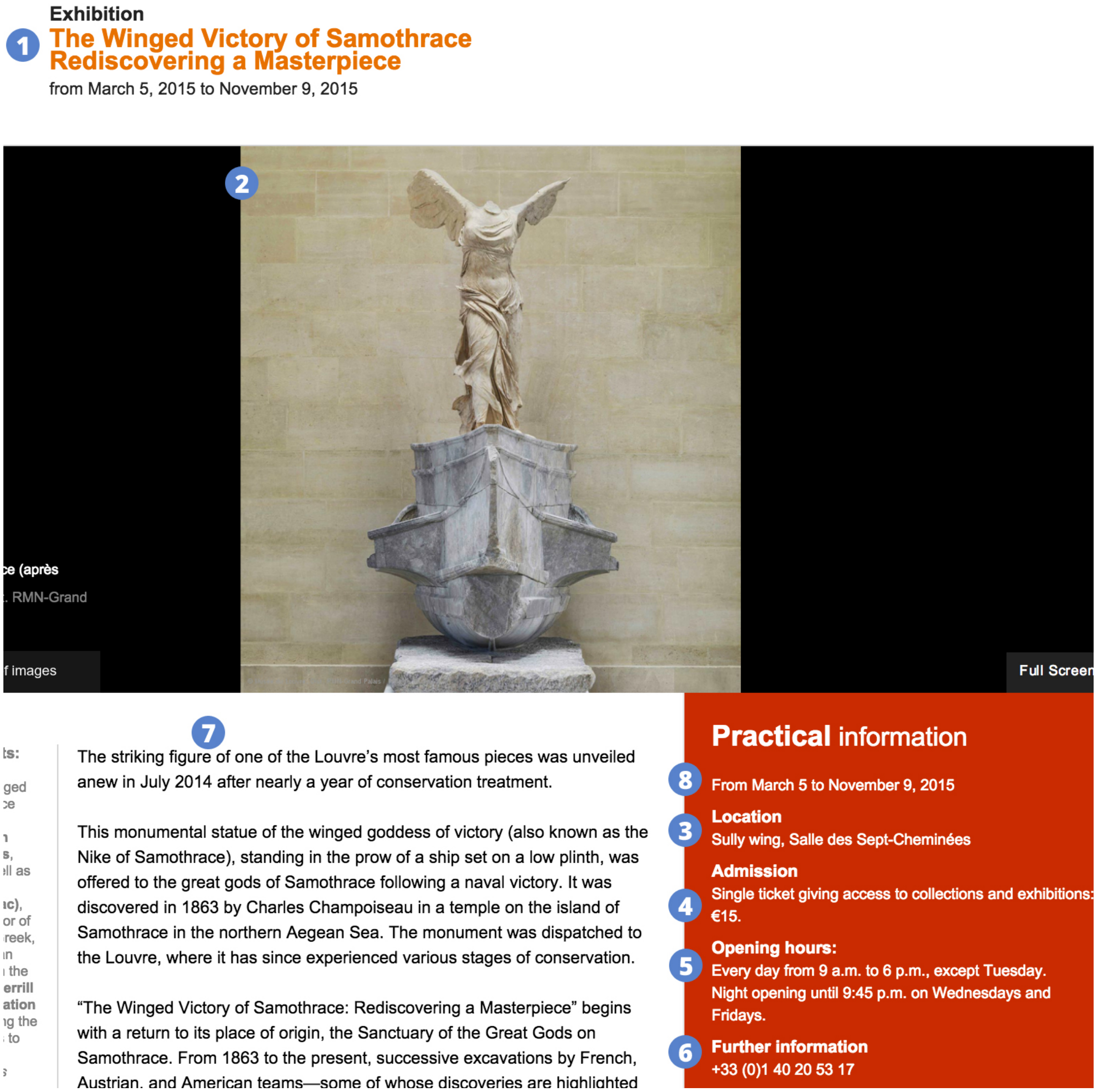
case, the pattern knowledge base and the static enrichment (multimedia, HTML structure) have a significant impact on the final result vocabulary.

7.3 Museum page

This use case is derived from the cultural heritage domain, which refers a great amount of Web entities that would add substantial value to the machine understandable Web if they were semantically annotated. The impact of our cultural heritage to the future of humanity is undeniabe. In this respect, lately the Europeana collections project[4] aimed to document the various artifacts of the European cultural heritage in one single directory, which at the moment accounts more than 53 milion entities. In the scope of the use cases for the proposed research, a webpage from one of the most popular statues of the Louvre museum, i.e. the Winged Victory of Samothrace, has been selected. Figure 7-3 depicts the webpage together with some labels that refer to content that could be annotated using vocabulary terms.

Studying the eight different information points of Table 7.5, the reader can realise the impact that this information could have if it was machine understandable. There are a lot of questions that could be answered by a search engine in a very accurate way if the webpage was crawled by exploiting the structured content. A list of the supported search questions could be the following:

[4] European collections: `http://www.europeana.eu/`



Exhibition

The Winged Victory of Samothrace
Rediscovering a Masterpiece

from March 5, 2015 to November 9, 2015

The striking figure of one of the Louvre's most famous pieces was unveiled anew in July 2014 after nearly a year of conservation treatment.

This monumental statue of the winged goddess of victory (also known as the Nike of Samothrace), standing in the prow of a ship set on a low plinth, was offered to the great gods of Samothrace following a naval victory. It was discovered in 1863 by Charles Champoiseau in a temple on the island of Samothrace in the northern Aegean Sea. The monument was dispatched to the Louvre, where it has since experienced various stages of conservation.

"The Winged Victory of Samothrace: Rediscovering a Masterpiece" begins with a return to its place of origin, the Sanctuary of the Great Gods on Samothrace. From 1863 to the present, successive excavations by French, Austrian, and American teams—some of whose discoveries are highlighted

Practical information

From March 5 to November 9, 2015

Location
Sully wing, Salle des Sept-Cheminées

Admission
Single ticket giving access to collections and exhibitions: €15.

Opening hours:
Every day from 9 a.m. to 6 p.m., except Tuesday.
Night opening until 9:45 p.m. on Wednesdays and Fridays.

Further information
+33 (0)1 40 20 53 17

Figure 7-3: Screenshot from a Louvre exhibition page with annotation placeholder labels.

- What is the entrance fee to visit the exhibition?
- Which is the faster path to the exhibition when I enter the Louvre museum?
- How does the monument look like?
- When can we visit the exhibition?
- Is the exhibition open at 17:00 tomorrow?
- When does the special exhibition about the Winged Victory of Samothrace end?
- What should I know for the exhibition?
- Could I have a short description of the exhibition?

The above listed questions are only a few that could be answered by having the webpage information annotated with semantics, as the agent that would need to answer would be able to understand what the various data values of the webpage refer to. Of course, the context of the user is an important factor, which includes the issue date and time of the question and the place at least. In conjunction with the webpage information, an agent could answer with a very accurate response to the 2^{nd}, 4^{th}, 5^{th} and 6^{th} question. For the rest of the questions, the accuracy of the answer is expected to be even better as the only input needed is the annotated data of the webpage.

Similarly to the previous use case, Table 7.6, summarises the result vocabulary for the input webpage of the museum exhibition use case. Most of the keywords are matched to the corresponding vocabulary term based on the text similarity and scoring of the corresponding candidate vocabulary terms. However, the term that will represent the contact information, i.e. the phone number, of the exhibition is based on the format of the presented data. The phone number on the webpage matches the phone pattern that the framework is able to recognise (based on the standards E.123[5] and E.164[6]). Furthermore, the picture annotation is following the multimedia types

[5] Phone number standard E.123: `https://en.wikipedia.org/wiki/E.123`

[6] Phone number E.164 standard: `https://en.wikipedia.org/wiki/E.164`

Keyword	Result vocabulary term	Comment	Information
Exhibition	schema.org:ExhibitionEvent		1
Location	swpo:Location		3
Ticket	schema:Ticket		4
Opening hours	gr:OpeningHoursSpecification		5
Statue	dbpedia-owl:Monument		7
-	schema:telephone	Pattern based	6
-	schema:Date	Pattern based	8
-	schema:ImageObject	Multimedia term	2
-	schema:name	HTML H1	1

Table 7.6: Result vocabulary of the museum webpage use case. The last column corresponds to the label number of the information that is expected to be annotated as described in Table 7.5 and in Figure 7-3.

recommendations, while the rest of the keywords are addressed within five different vocabularies, including the GoodRelations ontology (gr:), Schema.org (schema:), the Semantic Web Portal ontology (swpo:), the DBpedia ontology (dbpedia-owl:), and the Friend of a Friend vocabulary (foaf:).

7.4 Article page

The last use case is derived from the news domain, and specifically it refers to a scientific news article. Although the selected article is scientific, the findings can be generalised to the majority of article types. The selected article refers to some news published by the Jet Propulsion Laboratory of NASA, and discusses the latest findings of the Curiosity's Rover team that prove the existence of ancient lakes on the surface of planet Mars. The article includes a list of pictures of the planet's rocky surface, and a detailed explanation of the informational points that the assumption of the article's title is based on. Figure 7-4 depicts the webpage together with some labels that refer to the main parts of the content. These parts could be annotated using vocabulary terms in order to transform the webpage to a machine understandable knowledge resource. As shown in the figure, the main article includes also related links that are very important for the article's presentation. Finally, the publication date and the author of the article are important assets of any article and should be

Label number	**Information**
1	Title
2	Publication date
3	Article body
4	Link
5	Author
6	Modification date
7	Image

Table 7.7: Analysis of the various information bits on the NASA's article webpage that can be annotated using the appropriate vocabulary terms.

Keyword	**Result vocabulary term**	**Comment**	**Information**
News	bibo:Newspaper		3
Link	sioc:link		4
Updated	dbpedia-owl:updated		6
-	schema:email	Pattern based	5
-	schema:Date	Pattern based	2,6
-	schema:ImageObject	Multimedia term	7
-	schema:name	HTML H1	1

Table 7.8: Result vocabulary of the article webpage use case. The last column corresponds to the label number of the information that is expected to be annotated as described in Table 7.7 and in Figure 7-4.

annotated.

- Are there any new findings about planet Mars?

- Is there water on Mars?

- Who is in charge of the NASA Rover team news post?

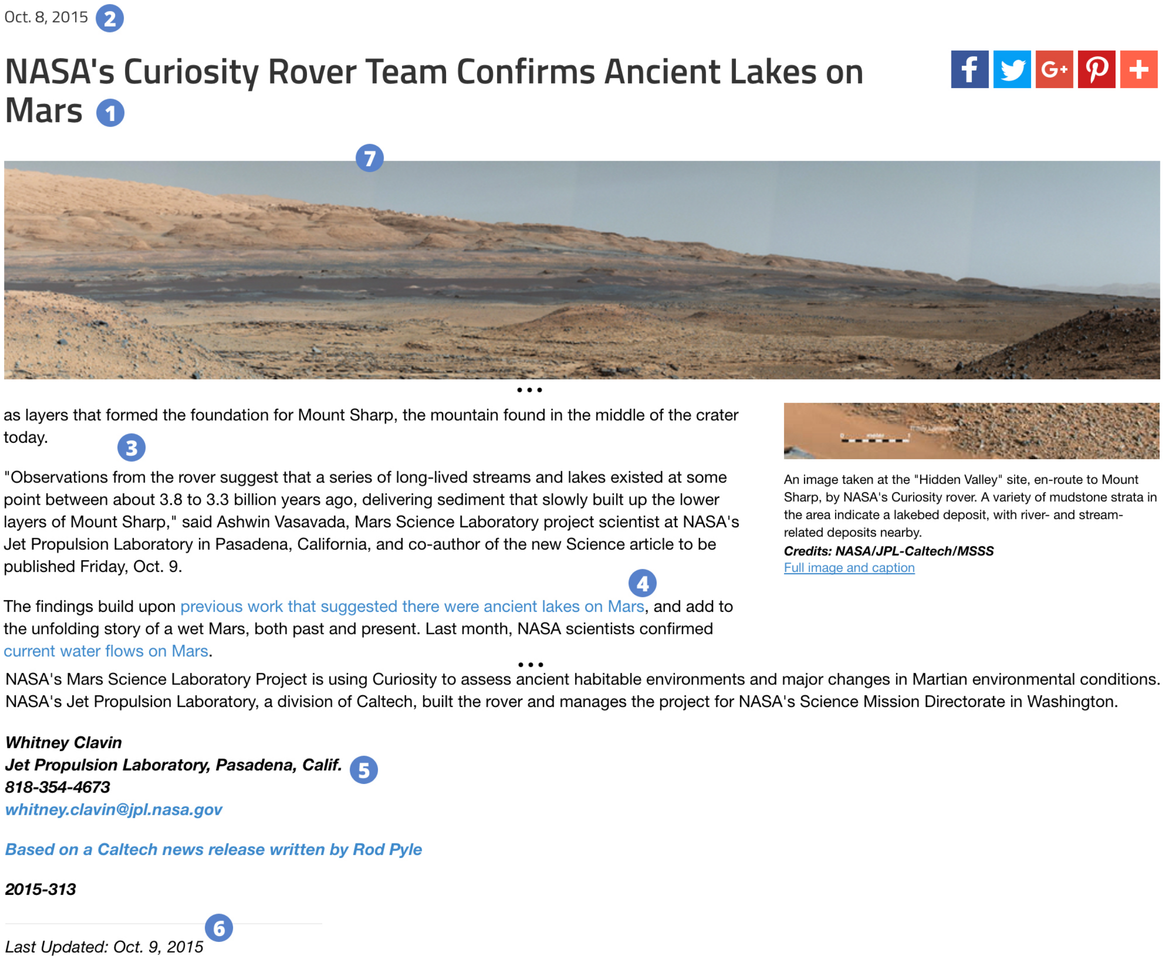
- When can we visit the exhibition?

- Is there any other related article?

- May I see how Mars surface look like?

- Do we have any image of Mars' lakes?

- Did NASA have any news about Mars in 2015?



Oct. 8, 2015

NASA's Curiosity Rover Team Confirms Ancient Lakes on Mars

• • •

as layers that formed the foundation for Mount Sharp, the mountain found in the middle of the crater today.

"Observations from the rover suggest that a series of long-lived streams and lakes existed at some point between about 3.8 to 3.3 billion years ago, delivering sediment that slowly built up the lower layers of Mount Sharp," said Ashwin Vasavada, Mars Science Laboratory project scientist at NASA's Jet Propulsion Laboratory in Pasadena, California, and co-author of the new Science article to be published Friday, Oct. 9.

The findings build upon previous work that suggested there were ancient lakes on Mars, and add to the unfolding story of a wet Mars, both past and present. Last month, NASA scientists confirmed current water flows on Mars.

An image taken at the "Hidden Valley" site, en-route to Mount Sharp, by NASA's Curiosity rover. A variety of mudstone strata in the area indicate a lakebed deposit, with river- and stream-related deposits nearby.
Credits: NASA/JPL-Caltech/MSSS
Full image and caption

• • •

NASA's Mars Science Laboratory Project is using Curiosity to assess ancient habitable environments and major changes in Martian environmental conditions. NASA's Jet Propulsion Laboratory, a division of Caltech, built the rover and manages the project for NASA's Science Mission Directorate in Washington.

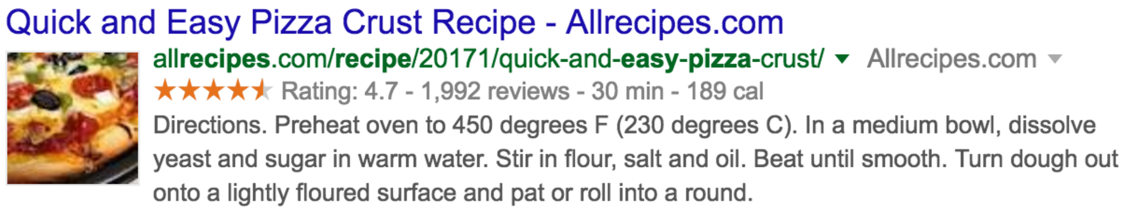
Whitney Clavin
Jet Propulsion Laboratory, Pasadena, Calif.
818-354-4673
whitney.clavin@jpl.nasa.gov

Based on a Caltech news release written by Rod Pyle

2015-313

Last Updated: Oct. 9, 2015

Figure 7-4: Screenshot from a NASA article about the exploration of Mars. The figure contains three parts of the webpage combined together.



Quick and Easy Pizza Crust Recipe - Allrecipes.com

allrecipes.com/recipe/20171/quick-and-easy-pizza-crust/ ▾ Allrecipes.com ▾

★★★★★ Rating: 4.7 - 1,992 reviews - 30 min - 189 cal

Directions. Preheat oven to 450 degrees F (230 degrees C). In a medium bowl, dissolve yeast and sugar in warm water. Stir in flour, salt and oil. Beat until smooth. Turn dough out onto a lightly floured surface and pat or roll into a round.

Figure 7-5: Screenshot from the Google search results for the keywords *easy pizza recipe*.

According to Table 7.8, the various comments that accompany the retrieved vocabulary terms indicate the difficulty of suggesting terms that reflect the semantics of the context from where a keyword has been extracted. *Date* and *Email* keywords do not appear in the text of the webpage, but they are derived from their datatype that is found within the textual content (i.e. the "Oct 8, 2015" and the email address accordingly). Furthermore, the first recommendation which is the term *bibo:Newspaper*, is a bit different than what the article is for, but still it can be considered a good starting point to explore related terms.

7.5 Summary

The selection of the use cases was mostly based on an attempt to cover a few domains with specific data points in the corresponding entities of it. For example a hotel room webpage represents the local business domain, while it could also be the page of a restaurant. The recipe domain was selected as it is one of the first domains for which the Google search engine employed the understanding of structured data. As shown in Figure 7-5 for the keywords "easy pizza recipe", the result presentation does not only include an excerpt of the webpage, but also the review score together with the number of reviews, the preparation time and the calories. All this data is extracted by the Google crawler by seeking for the corresponding semantic annotations.

The museum webpage represents the culture heritage domain, which is considered an important source of information that could be useful to have a structured and semantic based representation. Finally, the NASA's article follows the typical structure of an informational or news article, which is also a domain that could be really useful to become accessible in a machine understandable way.

Leveraging all the above domains to semantically rich representations facilitates the usage of the corresponding data by an agent in order to answer user requests, but also allows the better cooperation between various stakeholders without the need of an additional API. It is worth mentioning that the presented approach is not limited to these four domains and none of the design decisions of the algorithm has been created for a specific domain or use case. Other domains that could be tested against the methodology include the product catalog of an enterprise, e-shops, etc. If a company with a product catalog could provide a structured representation of the products using semantics, then the various partners could directly consume those pages as all the product characteristics would be explicitly specified and would be safe to gather the respective information directly from the page in an automated way (crawler), by being able to search for specific structures and values accompanying relevant vocabulary terms.

Finally, a methodology like the presented one that was used against the four above described use cases, is a generalised one without having any aspect that would help to customise it for a given domain. However, providing the possibility to train the underlying system to react differently for each domain, it would help to also provide domain specific recommendations for vocabulary terms.

Chapter 8

Evaluation

Measuring the effectiveness of the approach

Designing a methodology that it automatically performs a set of steps, brings many benefits to the applied field. The main drawback of automation over manual processes is the quality of the performed task. However, even in cases that the quality is considerably lower, it is still valuable to introduce automation and substitute a part of the respective workflow. This being said from an abstract point of view, the case of the discovery of vocabulary terms will be evaluated against the corresponding manual approach in order to realise the benefits and drawbacks.

In this scope, a set of dimensions and criteria have been considered for the evaluation of the presented approach. The plain manually performed keyword search discovery of vocabulary terms presented in the survey of Section 3.3 highlighted a few issues that the the proposed approach aims to address. Firstly, the duration of the vocabulary terms selection process across all the use cases lasted for not less than one hour in average, while in a lot of the participants submissions, we observed, as shown in Figure 3-5, very high values that go beyond the two hours. The discovery speed dimension is only one of those that the approach should be evaluated against in order to draw insights about the efficiency and effectiveness of it.

The presented chapter provides the evaluation details starting from the definition of the criteria that are used to judge the approach and the initial assumption that

Dimension	**Metric**	**Measurement**
Speed	$ET_{\mathcal{T}}$	Term discovery elapsed time
Accuracy	*precision*	Percentage of result terms that are relevant
Comprehensiveness	*recall*	Percentage of result terms that are retrieved

Table 8.1: Overview of the various evaluation dimensions and the respective metrics.

the automatic discovery of vocabulary terms can be up to 80% as accurate as the manual process by a human that has little or no experience in the subject of semantic annotations and the vocabularies role in that. Therefore, the criteria are defined in Section 8.1, the human based evaluation is described in Section 8.2 and the machine based one that can be performed in large scale is defined in Section 8.3.

8.1 Evaluation criteria

Putting the different aspects together, a set of criteria has been introduced to evaluate the designed methodology. In the information retrieval discipline, there are two widely used metrics that facilitate the measurement of the effectiveness of a retrieval approach, i.e. precision and recall. In the context of the LOVR approach, these two metrics are adopted accordingly as presented in Definition 29 and Definition 30. Beyond those two metrics, Table 8.1 provides a quick overview of the whole set of criteria employed together with a short explanation of their purpose. Following the short description of them, the various definitions, presented later in the section, elaborate on the formula, aim and usefulness of the metric.

Although all the above-mentioned are quantitative metrics, there are many scenarios that would be acceptable and the various metrics should be examined all together in order to be able to judge the approach accounting all the perspectives and trade offs.

The **speed** dimension is an important factor when a new methodology or approach is proposed. It is important to compare the time needed for the proposed approach to provide the same result with the existing workflows in order to gather insights about the contribution of the approach in the efficiency of the current solutions.

However, as the approach is automated it is not expected to produce the exact same result. The speed criterion can only positively affect the overall picture, as the elapsed time is significantly lower in comparison to a manual discovery process. Therefore, what is crucial to examine is the accuracy and comprehensiveness of the approach as discussed in the following definitions. The formula for the speed metric is presented by Definition 28.

Definition 28 (Vocabulary terms recommendation speed rate) *Let ET_k be the elapsed time for the output vocabulary term of the framework for a keyword k of the target webpage w, then the elapsed time of the discovery process per term is defined as the sum of all the ET_k values for the set of the result terms divided by the size $N = |\mathcal{T}|$ of the result vocabulary $\mathcal{T}$:*

$$ET_{\mathcal{T}}(w) = \frac{\sum_{i=1}^{N} ET_{k_i}}{N}$$

Precision captures the effectiveness of the approach to mostly retrieve results that are relevant, even if relevant results are missing. Thus, precision is the fraction of retrieved documents that are relevant. The relevant retrieved documents are considered to be the true positives, while the non relevant are the false positives. Respectively, the false negatives are the relevant ones that were not retrieved and the true negatives are the non relevant that were not retrieved as well. Therefore, in order to calculate the precision of the vocabulary recommendations approach, we need to count the true positives, which are the terms that appear at the output of the approach, i.e. $t \in \mathcal{T}$, and are also part of the terms that are considered relevant. Depending on the evaluation scenario, the set of relevant terms could refer to another source in order to accommodate the purpose of it.

Definition 29 (Vocabulary terms recommendation precision) *Let RT_w be the vocabulary terms set that is relevant for a target webpage w, then the precision is defined as the cardinality of the intersection set of the relevant terms with the discovered* *terms T_w divided by the number of the discovered vocabulary terms:*

$$precision(w) = \frac{|RT_w \cap T_w|}{|T_w|}$$

Recall captures the effectiveness of the approach to retrieve the total amount of documents that are relevant no matter if it retrieves non relevant as well. Therefore, in order to calculate the recall of the vocabulary discovery approach, we count the true positives and divide them by the total number of relevant terms.

Definition 30 (Vocabulary terms recommendation recall) *Let RT_w be the vocabulary terms set that is relevant for a target webpage w, then the recall is defined as the cardinality of the intersection set of the relevant terms with the total retrieved terms divided by the number of the relevant vocabulary terms:*

$$recall(w) = \frac{|RT_w \cap T_w|}{|RT_w|}$$

F-measure / F_1 is a measure that combines precision and recall by providing the harmonic mean[1] of them. The most used F-measure is the balanced F-score, which is the result of the division of the square of the geometric mean with the arithmetic mean. This is also called F_1 due to the case that the factor β at the general F_β-measure equation has value $\beta = 1$.

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{8.1}$$

F_β**-measure** / F_2 is the more generic measure that the previous one derives from when $\beta = 2$. F_2 gives a higher weight to the recall score over the precision. In the vocabulary terms discover, it is very important to gather all those terms that

[1] Harmonic mean Wikipedia: `https://en.wikipedia.org/wiki/Harmonic_mean`

would soundly describe the Web content. Thus, by employing the F_2 measure, the comprehensiveness of the approach is consider more important than the accuracy.

$$F_\beta = (1+\beta^2) \cdot \frac{precision \cdot recall}{\beta^2 \cdot precision + recall}, \beta = 2 \qquad (8.2)$$

The evaluation of the proposed approach has been conducted in both an automatic (machine-based) and a manual way (human-based). The various metrics that are used throughout both scenarios have been explained in this section.

8.2 Human-based evaluation

Evaluating the thesis methodology by employing human evaluators has both advantages and disadvantages. The main advantage is the knowledge of the individuals that can be used to leverage the whole process to a great feedback loop via the "wisdom of the crowd". On the other hand the main drawback is the possible non-deterministic interpretation of a problem by an evaluator due to contextual factors that cannot be controlled by the evaluation setup. Therefore, we consider the two evaluation approaches (machine and human based) complementary and both serving in their way the need of comprehensively evaluating the proposed algorithms.

The workflow of the approach starts with the distribution of an assignment to users, which asks to annotate specific items of information following the manual terms discovery process, which does not include the usage of the proposed approach. The same items were given to the methodology of the framework and afterwards the results are compared based on the four criteria of the evaluation approach presented in Section 8.1. The questions that were given to the participants are shown in detail under Appendix A. Each one of the evaluators is given one of the four use cases mentioned in Appendix A, and they are asked to answer a few questions related to vocabulary terms discovery by using the LOV search feature. The reported results show that the task is completed from the majority of the participants with uncertainty about the final outcome.

Apart from the questionary, a major dimension of the evaluation refers to the

Use Case	**Expertise level**						**Total**
	0	1	2	3	4	5	
hotel	11	5	1	0	0	0	17
museum	14	5	0	0	0	0	19
article	10	4	0	1	0	0	15
recipe	9	3	1	0	0	0	13
Total	44	17	2	1	0	0	64

Table 8.2: Overview of the participants demographics. The expertise level 0 refers to a participant with the limited knowledge of the Semantic Annotations topic, while level 5 refers to an expert in Semantic Web.

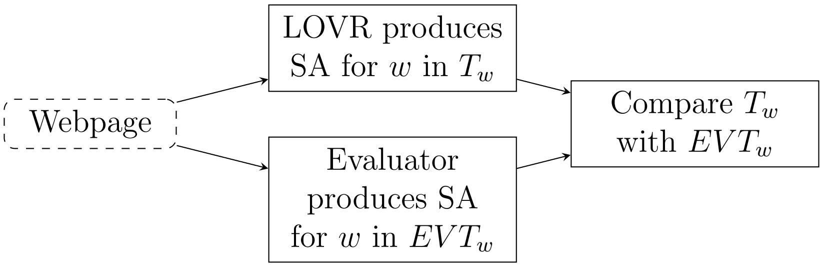

Webpage
LOVR produces SA for w in T_w
Evaluator produces SA for w in EVT_w
Compare T_w with EVT_w

Figure 8-1: The evaluation plan based on vocabulary terms discovery by evaluators. In the diagram *SA* stands for *semantic annotation*, w refers to the webpage, EVT_w represents the set of vocabulary terms by evaluators about the webpage, and T_w refers to the set of vocabulary terms that was generated by running the LOVR approach for the webpage w.

demographics of the participants. Table 8.2 provides an overview of the distribution of the participants across the evaluated use cases and their expertise level. As it is depicted in this overview, there distribution of the participants across the various use case is balanced, with an average (mean and median) of 16 participants per use case. However, the expertise level is not that uniformly distributed across the pool of participants, due to the fact that all of them are students at the University of Innsbruck and participate in the Semantic Web related course but without any priori experience to it.

The evaluators-based approach demonstrated in Figure 8-1 shows how the produced set of vocabulary terms for a given webpage w by an evaluator, i.e. EVT_w is used against the produced set of terms by the LOVR approach, i.e. T_w, in order to measure the effectiveness of the latter. As the activity diagram shows, for each evaluator, a set of metrics is computed against T_w. In this way, we can later measure up to which percentage the LOVR approach meets the evaluators base and also if it outperforms the results by the evaluators. Thus, EVT_w is the set of suggested terms by the evaluators, as shown in equation 8.3.

$$EVT_w = \{t : t \in \mathbb{VT}, \text{suggested by the evaluator for the target document } w\} \quad (8.3)$$

The abstract equation that is used to calculate the average for the various metrics is presented by equation 8.4, and it differentiates between the various webpage cases that are given to the evaluators. Therefore, for each webpage we will have an average value for each of the metrics presented earlier in the introduction of the evaluation chapter. Let N be the number of evaluators of the given webpage case, then the metric mean is formulated as:

$$\overline{m}_w = \frac{1}{N} \cdot \sum_{i=1}^{N} m_w^i \quad (8.4)$$

The four use cases that were distributed to the evaluators are investigated separately as the difficulty level of the use cases varies. A difference was expected to be observed among the various use cases, with the recipe one to be the easiest as it contains very structured content by nature (i.e. ingredients, steps to follow, nutritional data, etc.). The set of tables 8.3, 8.4, 8.5, 8.6 present the comparison of the criteria metrics values between the participants of the evaluation and the approach.

The application details of the criteria and any related assumptions are presented in the following list:

- Speed: The elapsed time of the discovery process has been provided by the participants. The approach measurement has been made technically via the implementation of the algorithm.

- Precision: Measured by evaluating the relevance of the proposed terms.

- Recall: Measured by evaluating if the proposed terms reflect all the information that can be annotated.

The calculation of *precision* and *recall* is based on the assumption that the set of relevant terms RT_w as declared in Definition 29 is identical to the union set of proposed terms per use case by the participants. Therefore, to generate the scores for the criteria of the manual annotations, for each participant a score is calculated based on the proposed terms of that participant compared to the set of relevant terms, or in other words, compared to the combined set of terms proposed by the participants. Similarly, the approach is compared to that union set of relevant terms proposed by the participants in order to see how it competes. The precision and recall formulas are adjusted according to the equations 8.5, 8.6 and 8.7. Let ET_w be the union set of all the proposed vocabulary term sets EVT_w of the N participants, after having removed any irrelevant terms.

$$ET_w = \bigcup_{i=1}^{N} EVT_w(i) \tag{8.5}$$

$$precision(w) = \frac{|ET_w \cap T_w|}{|T_w|} \tag{8.6}$$

$$recall(w) = \frac{|ET_w \cap T_w|}{|ET_w|} \tag{8.7}$$

Recall evaluates if all the relevant terms have been proposed. Therefore, the approach is compared to the aggregated set of terms proposed by the evaluators in order to measure how close to that it performs. In addition to that set of terms, the pattern based ones and the static terms have been added when applicable, as it is not a subjective set of terms and keywords, but are based on the content of the webpage without human intervention. On the other hand, the precision metric evaluates the amount of proposed terms that are in the set of the relevant ones and penalises when irrelevant terms are in the result set.

The tables 8.3, 8.4, 8.5, 8.6 that reflect the evaluation results for each use case, separate the autonomous from the semi-autonomous (with keywords), in order to be

Metric	**Manual**	**Approach**	**Delta (Δ)**
Speed (min)	64.50	1.95	-62.55
Precision (%)	88.60	63.63	-24.97
Recall (%)	29.40	88.23	+58.83
F_1	0.44	0.74	+0.30
F_2	0.33	0.82	+0.49

Table 8.3: Human-based evaluation results for the article use case. The values refer to the means of the metrics.

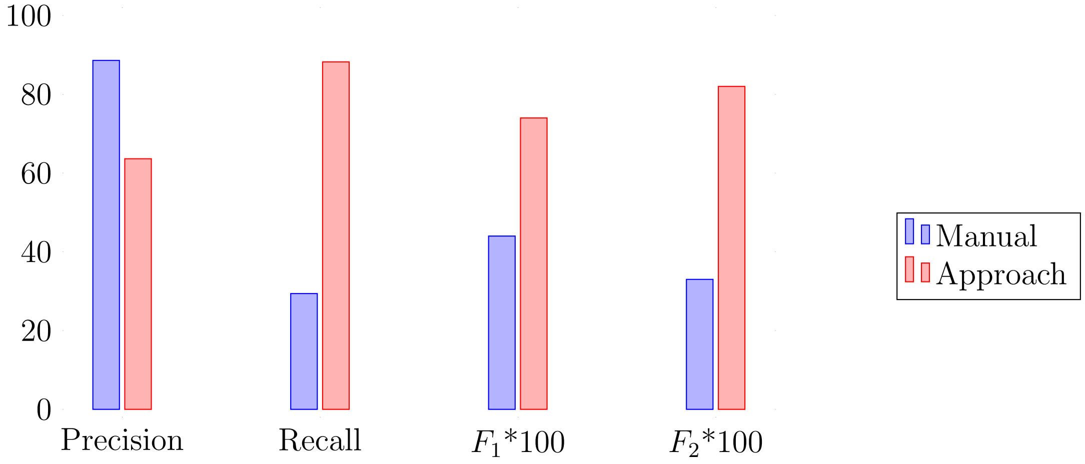


Figure 8-2: Recipe human-based evaluation comparison of metrics. The F-measures have been multiplied by 10^2 to have the same range with the percentages and be presentable in the same diagram.

able to compare them. The semi-autonomous refers to the usage scenario that the user explicitly defines the keywords to be used for the discovery process.

Furthermore, the participants mostly used the ranking of the terms as that is provided by the LOV search of terms, which ignores the usage of the authors as that was introduced in the presented approach in Definition 16 and aims to improve the score of vocabularies with potential. Improving the score of new vocabularies relies on the connections that vocabularies share through their authors and contributors.

Regarding the inclusion of static parts to the generated dataset we can easily observe the improved comprehensiveness of the proposed result terms in comparison to other approaches that ignore those information bits and to the survey result sets compiled by the participants. It could be argued that the recommendation of the static parts is not as important as the rest of the generated vocabulary, however, we find it

Metric	**Manual**	**Approach**	**Delta (Δ)**
Speed (min)	58.05	1.38	-56.66
Precision (%)	60.60	94.11	+33.51
Recall (%)	28.60	87.5	+58.90
F_1	0.38	0.91	+0.53
F_2	0.31	0.89	+0.58

Table 8.4: Human-based evaluation results for the hotel use case. The values refer to the means of the metrics.

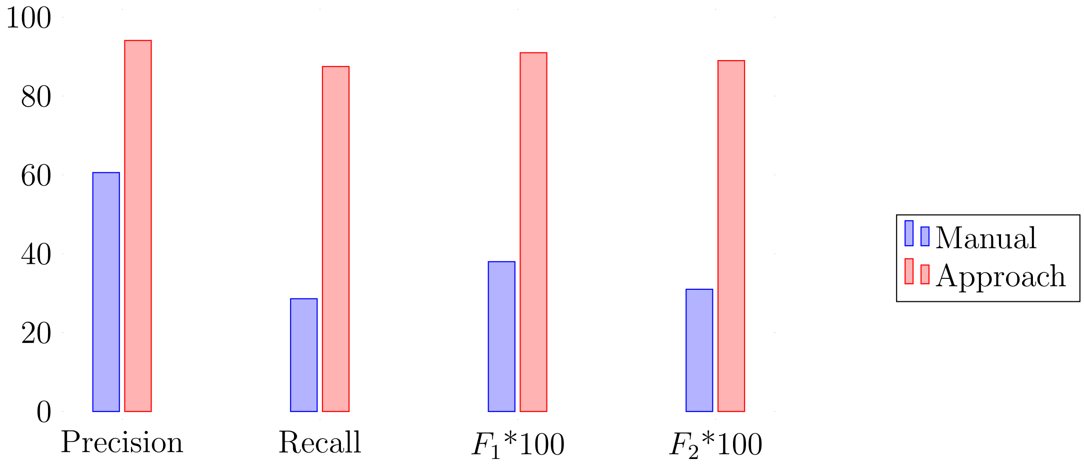


Figure 8-3: Hotel human-based evaluation comparison of metrics. The F-measures have been multiplied by 10^2 to have the same range with the percentages and be presentable in the same diagram.

Metric	**Manual**	**Approach**	**Delta (Δ)**
Speed (min)	43.88	0.98	-42.90
Precision (%)	60.80	72.72	+11.92
Recall (%)	23.10	90.90	+67.80
F_1	0.33	0.81	+0.48
F_2	0.26	0.86	+0.60

Table 8.5: Human-based evaluation results for the museum use case. The values refer to the means of the metrics.

Metric	**Manual**	**Approach**	**Delta (Δ)**
Speed (min)	40.15	0.75	-39.40
Precision (%)	66.30	90.90	+24.60
Recall (%)	25.80	83.83	+57.53
F_1	0.37	0.87	+0.50
F_2	0.30	0.85	+0.55

Table 8.6: Human-based evaluation results for the recipe use case. The values refer to the means of the metrics.

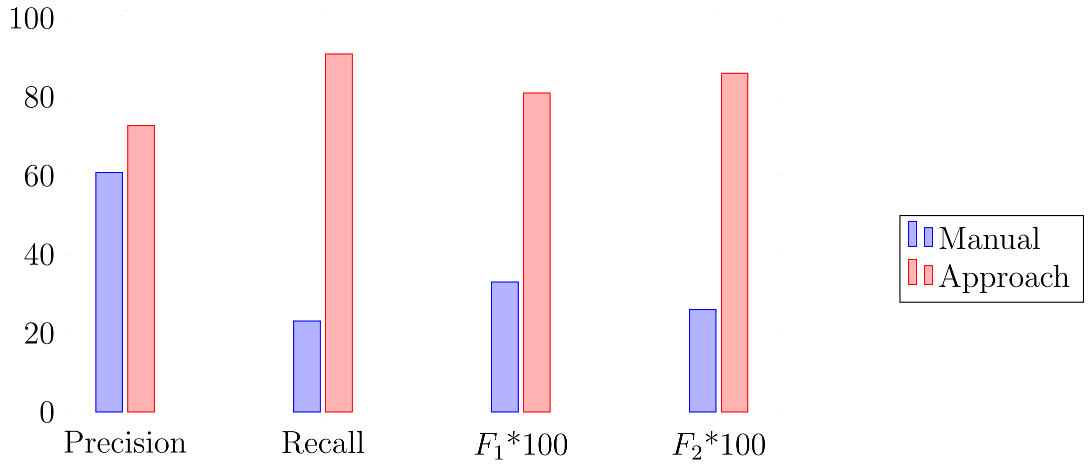


Figure 8-4: Museum human-based evaluation comparison of metrics. The F-measures have been multiplied by 10^2 to have the same range with the percentages and be presentable in the same diagram.

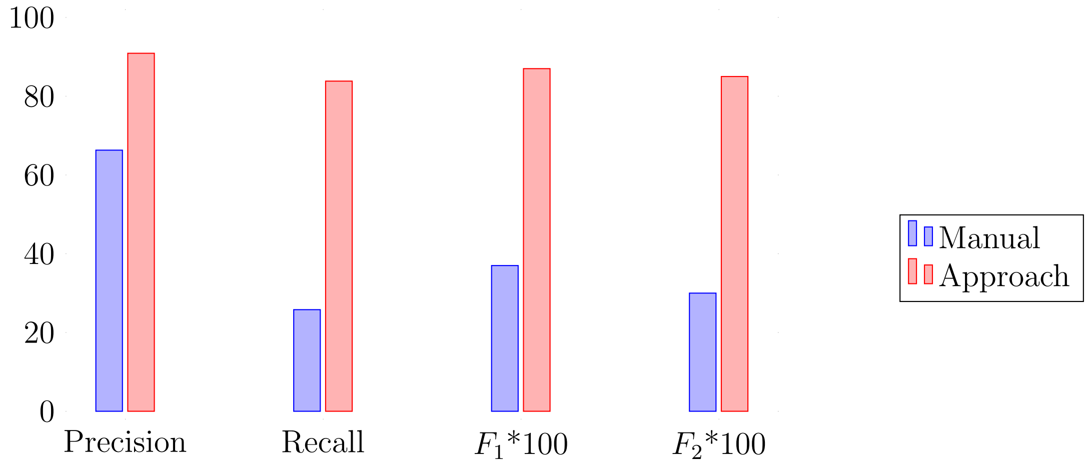


Figure 8-5: Recipe human-based evaluation comparison of metrics. The F-measures have been multiplied by 10^2 to have the same range with the percentages and be presentable in the same diagram.

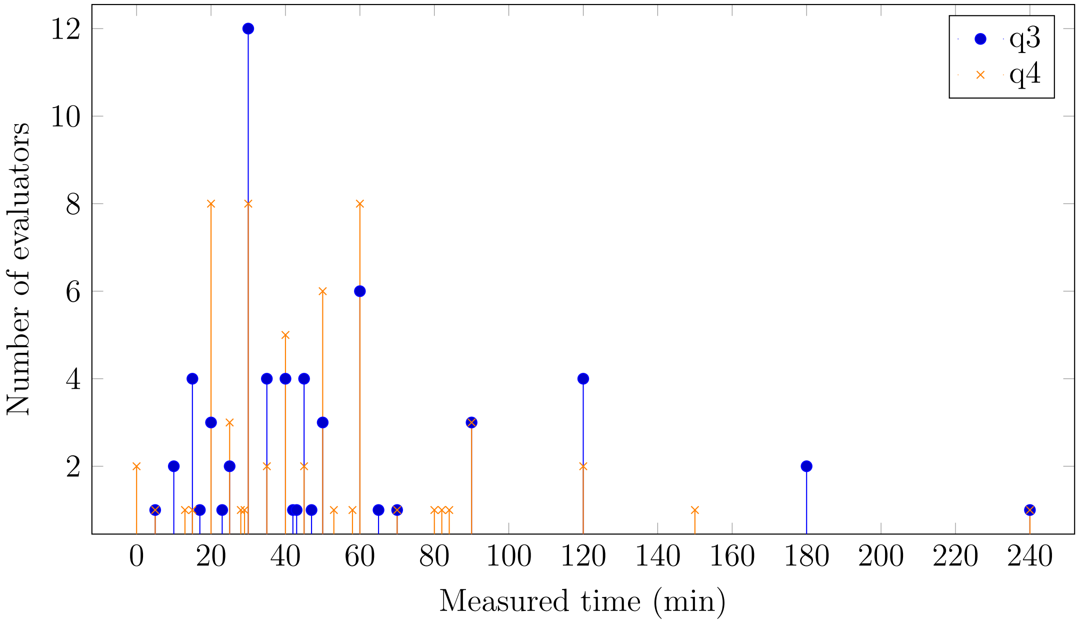


Figure 8-6: Histogram showing distribution of the time needed by the evaluators to select vocabulary terms (labeled as q3) and to build the JSON-LD (labeled as q4).

extremely useful for new vocabulary engineers or developers to be explicitly informed about the importance of annotating the multimedia content of the webpages. An indication of the impact of those terms in the LOVR approach is the results of the survey that we presented in Section 3.3, where none of the participants provided vocabulary terms and suggestions for annotating the multimedia content, which existed in all the use cases. Furthermore, underlying the importance of explicitly specifying the multimedia content on a webpage, we would like to stress out that the search engine providers not only crawl text from the Web sphere, but also images and videos and on top of the crawled data they provide the corresponding search functionality, which benefits and becomes more accurate when the multimedia content is explicitly annotated.

Finally, comparing the output result set of terms with each one of the individual collections compiled by the evaluators, we realise a significant difference between the selected terms by the participants and our approach. Table 8.7 refers to the recipe use case and shows that only a few participants selected the same terms with the approach algorithm. Some of the participants chose a different one from the top

Approach proposed term	Evaluation occurrences
schema:Recipe	15%
schema:ingredients	15%
schema:totalTime	7%
schema:prepTime	7%
schema:recipeYield	7%

Table 8.7: Comparison of the proposed approach results with the survey participants' input.

three candidates, e.g. for the *Recipe* class, while some other terms were wrong (e.g. the schema:CookAction) as such information cannot derive from the webpage.

8.3 Machine-based evaluation

The second type of evaluation that has been followed is designed in a way that benefits from webpages that already include semantic annotations. This evaluation methodology allows to compare the results of the proposed approach on webpages with the semantic annotations that those already have, but without taking them in consideration in the vocabulary terms discovery process. Therefore, there is no manual intervention to the evaluation, which helps to provide an evaluation approach without any possible bias.

For each one of the use case types that were discussed in Chapter 7, a few representative webpages with semantic annotations have been manually gathered into one pool of webpages that is given as input to the LOVR approach with an evaluation extension to the vocab-recommender implementation. The evaluation flow that is followed is depicted in Figure 8-7.

As depicted in the flowchart of Figure 8-7 the first step is to extract the semantic annotations that are present in the webpage under question. Those terms form the set of existing vocabulary terms EVT, as that is defined below by equation 8.8. The second step of the approach is to execute the recommendation process and generate the result vocabulary $\mathcal{T}$. Finally, the extracted set of terms with the generated vocabulary are compared with the evaluation criteria defined earlier in Section 8.1.

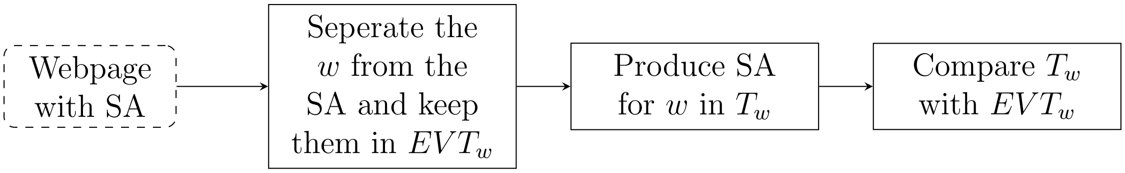

Webpage with SA
Seperate the w from the SA and keep them in EVT_w
Produce SA for w in T_w
Compare T_w with EVT_w

Figure 8-7: The evaluation flowchart based on webpages that already include semantic annotations. In the diagram SA stands for *semantic annotation*, w refers to the webpage, EVT_w represents the set of existing semantic annotations in the webpage, and T_w refers to the set of vocabulary terms that was generated by running the LOVR approach with input the webpage w.

In the previous evaluation approach, three different approaches were compared against the cumulative set of terms proposed (per use case) by the participants. This set played the role of the expected result vocabulary terms. Therefore, the precision and recall of the manual approach (survey), the semi-automatic approach and the automatic approach were all computed with the survey based expected result set as the set of relevant documents. In the scope of the machine-based evaluation, the criteria need to be slightly adjusted in order to reflect the purpose of it. The expected result set (relevant documents) is assumed to be the extracted vocabulary terms of each webpage, i.e. EVT_w.

$$EVT_w = \{t : t \in \mathbb{VT}, \text{appears in the target website document } w\} \tag{8.8}$$

$$precision(w) = \frac{|EVT_w \cap T_w|}{|T_w|} \tag{8.9}$$

$$recall(w) = \frac{|EVT_w \cap T_w|}{|EVT_w|} \tag{8.10}$$

Starting point for the execution of the above-mentioned evaluation plan is the specification of a list of webpages that will be used as input to the approach. The list of webpages has only one requirement, namely the inclusion of semantic annotations within the HTML source of the page in any valid format (JSON-LD, Microdata, RDFa Lite, etc.). In addition in order to be able to discuss the results of this evaluation with the results of the survey-based one, the list of webpages that has been selected

Metric	**Approach Keywords**	**Approach NLP**
Keywords	13	92
Speed (min)	1.20	7.30
Precision (%)	90	12
Recall (%)	65	65
F1	0.75	0.21
F2	0.68	0.34

Table 8.8: Machine-based evaluation results for recipe webpages. The values refer to the means of the metrics.

Metric	**Approach Keywords**	**Approach NLP**
Keywords	7.5	190.5
Speed (min)	0.81	17.10
Precision (%)	94	3
Recall (%)	71	71
F1	0.81	0.05
F2	0.74	0.13

Table 8.9: Machine-based evaluation results for article webpages. The values refer to the means of the metrics.

includes webpages from the same domains. The list of webpages and their type can be shown in Appendix B. The results after running the approach for that set of webpages are presented in the tables, Table 8.8 for recipe webpages, Table 8.9 for article webpages and Table 8.10 for local business webpages.

Table 8.8 presents the recipe use case metrics, which are very good in terms of the approach accuracy and a bit lower when it comes to recall. The reason is the rich content of the recipes and the various implied entities that were missed. For example the author of the recipe, the term that refers to the review body of the reviews or the review count related term. However, those review related terms are considered as technicalities that the webpage owner or developer would need to address within the development process and not terms that refer to content that was not interpreted properly by the approach. Thus, the approach proposes terms for the review but not properties around the class that will be needed during the implementation of the annotations.

Metric	**Approach Keywords**	**Approach NLP**
Keywords	14.5	462
Speed (min)	1.10	20.40
Precision (%)	58	1.82
Recall (%)	90	90
F_1	0.67	0.03
F_2	0.78	0.08

Table 8.10: Machine-based evaluation results for local business (hotel, restaurant) webpages. The values refer to the means of the metrics.

In the article use case, presented in Table 8.9, the recall scores are lower than the precision due to the fact that the extracted vocabulary terms (EVT) include references to the publisher of the article as an organisation entity. This information is only implied within the webpage, therefore missed by the approach. On the other hand, the approach finished with high accuracy, having only a few false positives. Finally, one of the two cases included one proposed term, i.e. *link*, that was not found in the EVT, but it would be correct to be added to the webpage.

Table 8.10 presents the evaluation of the proposed approach on the local business domain. The keyword based approach shows a strong performance in the *recall* metric, and the F_2 measure, which is translated to high comprehensiveness. The reason behind the lower performance in the accuracy dimension is due to the fact that the semantic annotations of the webpages are not that rich as those proposed by the approach. Manually checking the differences between the proposed set of terms with the set of extracted vocabulary terms, suffices to realise that there is space for improvement for the evaluated webpages and all the proposed terms would make sense to be included. For example, for the hotel webpage, the approach proposed to use the terms *schema:checkin, schema:checkout*, which is an important piece of information to provide to the different agents.

The above machine-based evaluation was conducted on the hypothesis that the pre-existing semantic annotations of webpages can be used as the evaluation base to compare the results of the approach and measure its effectiveness. The scores presented in the above tables include false negatives, which are result vocabulary

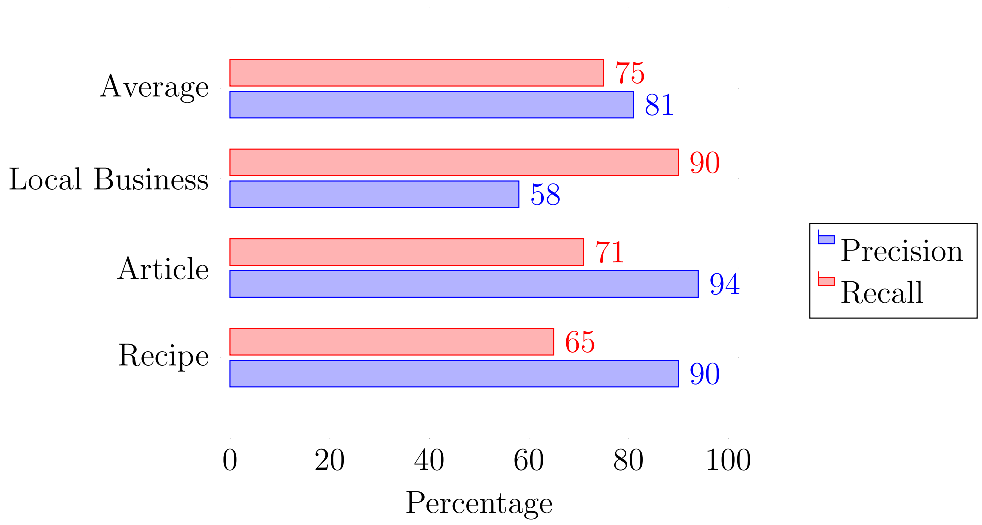


Figure 8-8: Precision and recall comparison among the evaluated domains.

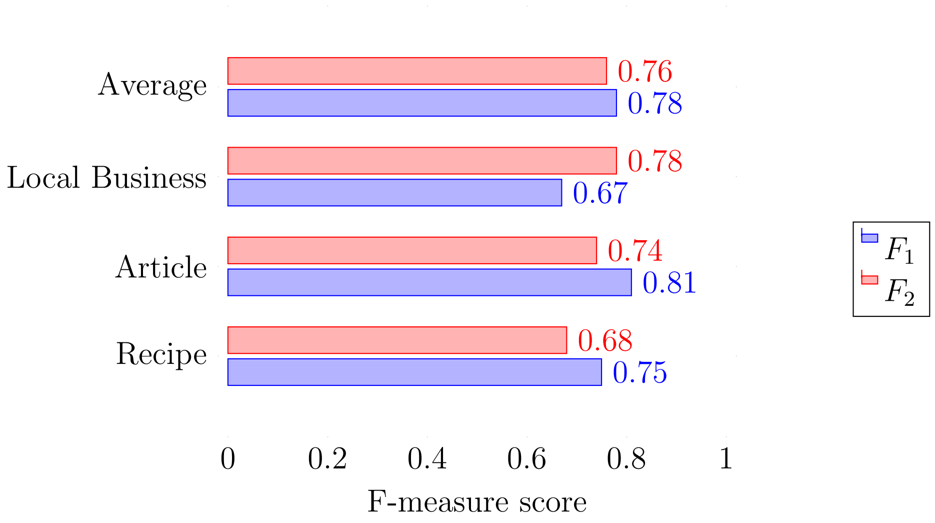


Figure 8-9: F-measure comparison among the evaluated domains.

Metric	**Approach Keywords**	**Approach NLP**
Precision (%)	80.66	5.60
Recall (%)	75.33	75.33
F_1	0.78	0.10
F_2	0.76	0.21

Table 8.11: Machine-based evaluation results for all the use cases together. The values refer to the means of the metrics.

terms that do not appear within the extracted vocabulary terms EVT. However, as it was already discussed those terms are considered relevant to the webpage under question. Furthermore, during the evaluation, it was interesting to realise the value of returning more than one but less than three terms as the result of a keyword. Three candidate terms allow the user to compare the three best options and select the one that better fulfils the requirements of the webpage.

Finally, calculating the average values across all the use cases about precision and recall, we see in Table 8.11 that the overall accuracy (precision) is above 80% while comprehensiveness (recall) is 75%. Those mean values are also depicted in the above mentioned figures that compare the three use cases. Regarding the NLP approach, the precision has a very low score, as the proposed approach did not focus in optimising the NLP extraction, but rather focused on the proposal of useful terms for a given set of keywords.

8.4 Summary

The human-based evaluation aimed to compare the results of the survey presented in Section 3.3 with the results of the approach when it is applied on the same webpages. In this scope, the evaluation examined two usage scenarios, i.e. a) using a manually defined keyword set to discover the corresponding vocabulary terms and b) by giving as input the webpage itself and relying on NLP for the extraction of the keywords. The comparison of the measurements for both scenarios with the manual annotations discovery is impressive and proves the benefits that were earlier formulated in the scope of the main hypothesis about the facilitation of the semantic annotations development. The speed of discovering the vocabulary terms was by far faster by the approach, as it would be expected for any automated process. In addition, the precision and recall of the approach clearly outperformed the participants of the evaluation as they were not experts neither in the field of semantic annotations nor the domain of the use cases. This fact highlights the need of employing a vocabulary discovery assistant in the process of annotating a webpage.

Furthermore, the machine-based evaluation aimed to compare the results of the approach on webpages that already have semantic annotations with the set of semantic annotations of the webpage itself. The main drawback of the human-based evaluation plan was the demographics of the participants, as they could not be considered as experts in the field of the Semantic Web. Therefore, this approach should complement the overall evaluation verdict. In order to do so, the workflow included the separation of the annotations from the content to create the set of expected terms, which will be used to evaluate the performance of the approach. The approach algorithms are not biased by any preexisting annotations, as they are oblivious to any semantic annotations of the webpage. The speed metric in this context is not of significant importance as the time that the developer of the webpage needed to develop the semantic annotations is unknown and cannot be compared to the process time of the approach. On the other hand, precision and recall provide a strong evaluation point of the approach as they show the effectiveness of it.

The evaluation results from both scenarios show the strong potential of the approach for being a first step in the process of discovering vocabulary terms by Web developers and website owners. The human-based evaluation showcased how the approach outperforms the evaluators results in terms of processing time, but also from the perspective of effectiveness given the better precision, recall and F-measure scores. The hypothesis at the beginning of this chapter, which is also under question within the whole thesis, has been proved through this evaluation, as the proposed approach achieved way more than 80% accuracy in the result set compared to the evaluators' results. Furthermore, it provides promising performance against the set of relevant terms for a webpage as that is defined by an expert, based on the findings of the machine-based evaluation, which shows a strong accuracy of 80% based on the simple keywords approach enhanced with patterns and static parts. On the other hand going from semi-automatic to full automatic seems to be a big difference, as the NLP approach is gathering too many keywords to use for vocabulary terms discovery afterwards. This fact has a negative impact to precision, while recall ideally remains the same as the extracted keywords is a super set of the manually selected ones. The usage of the patterns and static parts extraction is the same across the two methodologies.

Chapter 9

Conclusions and Future work

Beyond the vocabulary terms discovery assistant

An overview summary of the research results and any related future work, together with a few visionary directions on top of the presented research are the subjects that are discussed throughout this chapter. First and foremost, when the author of this dissertation decided to pursue a PhD, he had always in mind the applicability and the impact of the involved research work to the current and upcoming Web trends. At the beginning of the PhD research, the main topics on the hype was Social Web, Linked Open Data and Big Data. In this direction the various contributions presented throughout Chapter 2 explored the space of Web data extraction, modelling in the Social Web and applications using Linked Open Data. The results of the first research endeavours helped to shape the main contribution of the PhD work, i.e. the facilitation of the vocabulary terms discovery.

The facilitation of the discovery of vocabulary terms for the webpage, that is under question, combines existing solutions and the presented research in order to provide a methodology that accepts a website as input and provides a result vocabulary at the output. In the scope of the research endeavours, a survey that was ran and aimed to test the ease of manually discovering vocabulary terms, provided valuable insights about the difficulties and challenges that an engineer needs to address within the semantic annotation development process. The proposed methodology proves that the discovery process can be automatised with a significant success rate. The effectiveness of the proposed methodology is further discussed in Section 9.1, which summarises the contributions presented throughout the dissertation. Any limitations that has been observed are enlisted within Section 9.2, which discusses the research results.

What is the next step for the Web? How does the presented approach fit to the next step of the Web technology? These questions form the backbone of the last two sections, i.e. Section 9.3 and Section 9.4 respectively.

9.1 Research results

The main aim and accomplishment of the presented approach is the generation of a result vocabulary for a given set of keywords. This task is not a trivial engineering task and requires significant amount of effort in order to be resolved. The designed approach employs various quantitative metrics to support the ranking of the various vocabularies and vocabulary terms with the ultimate goal the generation of a result vocabulary with the most relevant and useful terms for the given input.

At the beginning of the thesis, within the introductory Section 1.2, a set of research questions were defined together with a thesis that was the aim of the contributions to prove. The major goal of the research work is to provide a methodology that facilitates the discovery of vocabulary terms for a target webpage in order to be used for the generation of semantic annotations of the target webpage.

The first research question reflects the core of the presented research by asking how a webpage can be leveraged to a machine-understandable interface in a semi-automatic way. To answer this question, Chapter 5 describes the designed solution and the methodology that was followed in the presented research work. The research approach is defined in the frame of design science by declaring the problem, the objectives, the design and development, the demonstration, the evaluation and the communication of it via conference publications. The question is answered by defining a process that takes in consideration the target webpage, the vocabulary space, the

Linked Open Data and data patterns of the content in order to extract a representative set of vocabulary terms from the vocabulary space. These terms could be later be used by an engineer to generate the corresponding semantics annotations, which will enable a webpage to be machine-understandable.

The second research question refers to the Linked Open Data (LOD) context and how LOD can be leveraged to support the selection of vocabulary terms for a given webpage. In the scope of the presented research, LOD has been used within the main ranking formulas that assign a score to the various retrieved vocabulary terms as it was described in Chapter 4 and specifically in Section 4.5. The LOD data is accessed via aggregations and indices provided by other approaches, which the presented work benefits from by integrating them in the main algorithm. The main aspect that the algorithm is looking into refers to the usage of the candidate vocabulary terms within the LOD cloud.

The third research question refers to the comparison between the various candidate vocabulary terms for a given keyword. The above described approach supports searching for vocabulary terms and more than one results are returned for each keyword. Therefore, the results need to be sorted based on some scoring. The score calculation is performed within the various ranking steps of the approach. The dimensions that are taken in consideration are: a) the popularity of the parent vocabulary within the vocabulary space according to the incoming interlinks; b) the popularity of the terms within the LOD cloud based on two repositories; c) the popularity of the contributors of the parent vocabulary according to the score of other vocabularies that they have contributed to; and d) the relevance of the retrieved term to the keyword.

Having defined all the theoretical parts, conducted a survey on the manual development and discovery of vocabulary terms for a given webpage, designed the algorithms that are used within the approach and answered all the research questions one equally important topic is addressed within Chapter 8, namely the evaluation of the approach. In this respect, two methodologies were employed in order to evaluate the effectiveness and efficiency of the proposed approach; a) a human-based evaluation and b) a machine-based evaluation. The former aimed to compare the results of the survey presented in Section 3.3 with the results of the approach when it is applied on the same webpages. Two subtypes of methodologies are followed here; a) using a manually defined keyword set and b) giving as input the webpage itself and relying on NLP for the extraction of the keywords. It is important to highlight that the approach outperformed the results that the manual discovery generated not at the level of each participant individually but at the aggregated set of terms across all the participants' result sets. In addition, the speed of discovering the vocabulary terms was by far faster by the approach, as it would be expected for any automated process. The latter aimed to compare the results of the approach on webpages that already have semantic annotations with the set of semantic annotations of the webpage itself. The results were as impressive as with the previous approach, while the expected set of terms for the relevant documents could be considered more reliable than in the previous approach due to the fact that the list of webpages that was evaluation in the current approach are real world examples with semantic annotations aiming to leverage the content to semantically structured data and machine-interpretable information.

The hypothesis of the thesis, which refers to semi-automatically generate vocabulary term recommendations for a given webpage with a recall over 80%, has been proved through this evaluation, as the proposed approach achieved way more than 80% accuracy in the result set and it also outperformed the evaluators' results. Furthermore, it provides promising performance against the set of relevant terms for a webpage as that is defined by an expert, based on the findings of the machine-based evaluation, which shows a strong accuracy of 80% based on the simple keywords approach enhanced with patterns and static parts.

9.2 Discussion and limitations

Many extensions and improvements can be made to the approach in order to increase the performance scores and achieve better result vocabularies. With the current design the approach is neglecting the usage of any language apart from English.

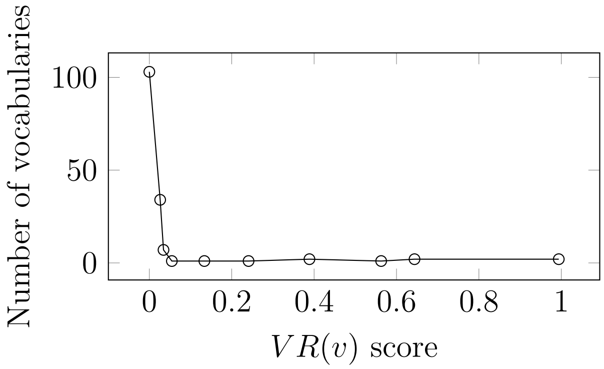


Figure 9-1: $VR(v)$ score frequency distribution (published in [79]).

For sake of simplicity, the approach was designed to work only in English, as all the vocabularies express their terms in English. However, it would be a great improvement to integrate a translation service, e.g. BabelNet [58] that would be used to translate the webpage content before extracting keywords.

In addition, it would be of great interest to experiment with synonyms of the extracted keywords and then compare all the candidate vocabulary terms to realise if within the top candidates there are terms that are coming from the synonyms and if those add value to the result vocabulary. For example, in the recipe use case most of the webpages use the wording *"Directions"* instead of *"Instructions"*, which is the term used within schema.org. This difference causes the approach to fail to retrieve the corresponding term.

One of the observations that was made during the research work presented above, is the fact that the overall highest ranked vocabularies may not cover concepts that are needed for a given domain. Using the LOV's SPARQL endpoint[1] to run the query shown in Listing 9.1, the frequency distribution of $VR(v)$ metric is calculated for the whole set of vocabularies as depicted in Figure 9-1. As the distribution diagram shows, most of the vocabularies score low between 0 and 0.1, while only a few make it beyond the 0.2 score, which could be considered a high value as it reflects a reusability of the vocabulary from the 20% of the registered vocabularies. Those that are very close to the maximum score value, are considered to be outliers as they are mostly

[1] LOV SPARQL endpoint: `http://lov.okfn.org/dataset/lov/sparql`

vocabularies that include very basic terms, like *rdf:*, *rdfs:*, etc. The introduction of contributors score within the ranking of vocabularies aimed to address this issue.

```
SELECT ?p ?b {
GRAPH <http://lov.okfn.org/dataset/lov>{
  ?vocab a voaf:Vocabulary.
  ?vocab vann:preferredNamespacePrefix ?p.
  ?vocab voaf:reusedByVocabularies ?b.
}} ORDER BY desc (?b)
```

Listing 9.1: LOV SPARQL for vocabularies reuse, as published in [79].

Summarising the outline of the approach, there are many more directions than the presented, that it can be extended to. Within the described contributions the aim is to address the discovery of terms, which is considered the first and foremost step towards the facilitation of the transformation of websites to machine understandable Web entities. After the discovery of the set of vocabulary terms that can support the implementation of semantic annotations of the given webpage content, the next step is to assist the generation of the semantic annotations by either providing examples, or by automatically providing an application proposal of the terms on the input content. The result could have high probability of being close enough to the desired quality as the vocabulary terms are connected with the corresponding keywords, which can be considered as the value of the various datatype properties that are recommended among the vocabulary terms. In addition, the terms have specific range, which can also be a helpful bit of information in the same direction.

Moreover, having generated structured entities for the given webpage, it would be possible to further annotate them with actions in order to fulfil the requirements of a basic API interface as that was described in Table 4.2. This direction has been examined in the scope of the presented research and published in [78] as described in Section 5.4.2. Deeper research in this topic could be conducted in order to create a complete transformation of a website to an API.

The presented approach, as it is depicted in Figure 5-1, Algorithm 1 and the dimensions presented in Section 5.4 about the result vocabulary enrichment, combines various external sources in order to gather usage patterns and quality indicators of the vocabulary terms in the vocabulary space. This information is leveraged to form a vocabulary that facilitates the production of annotations for a given webpage. The presented algorithm can be easily modified in order to follow different specifications, e.g. the generation of a result vocabulary with only the highest ranked terms.

On the other hand, it should be explicitly noted that the presented approach does not aim to automatically annotate the webpage with the suggested vocabulary terms. This direction is a different research topic, which would be similarly impactful to the presented methodology. The first step between the current approach status and the automatic annotation of a webpage, would be the generation of examples from the result vocabulary, that would allow an easier introduction of the user to the semantic annotations topic.

9.3 Future work

Moving forward and beyond the accomplishments of the presented approach there are two main directions to work on, a) improve the approach by adding more functionality or details to it and b) build new approaches on top of it in order to make one more step towards the realisation of the machine interpretable Web.

Firstly, the weaknesses of the approach presented in Section 9.2 is the motivation for the next step. The current design neglects the relationships of the proposed terms and constructs the result vocabulary by combining all of them together in a new set. However, taking in consideration the domain and range of the various datatype and object properties that the vocabulary terms represent, it would be possible to construct a result vocabulary that has better consistency and it would be easier to apply on a target webpage as the various recommended terms will be interconnected.

Furthermore, a slightly different implementation of the approach would allow the user to specify the set of vocabularies that she would like to explore and discover terms from. This functionality would be very helpful for those cases that the user would like to use terms only from schema.org, which is the standard vocabulary for a few domains, like the recipe domain.

Exploring relationships of the discovered terms within LOD and LOV could be another direction of research on top of the presented approach. The relationships between a discovered term with other terms that are met often together within the LOD could lead to even more relevant recommendations within the result vocabulary.

From another perspective, the approach could benefit from any existing annotations on the webpage in order to understand the context better and steer the recommendations accordingly via different scores in the ranking process. In addition keywords extracted from the most important elements of the webpage like HTML H1, H2, H3, H4 could be considered as good sources for a summary of what the webpage is referring to. Those keywords could be used to decide in which domain the webpage belongs to within a predefined list of domains. For example a webpage that in a heading of type H1 has the keyword hotel, it is highly probable that the rest of the webpage included content that would be possible to be annotated using vocabulary terms connected to the main term that represents a hotel entity.

In addition to the result vocabulary, it would be very helpful for the user of the approach to be provided with examples of the proposed terms in use. Extending the usage of Natural Language Processing and Named Entity Recognition technics it could be possible to provide examples of the proposed vocabulary terms used within the content of the target webpage.

A recent W3C recommendation, namely the Web Annotation Vocabulary[2] facilitates the description of annotations in a target webpage, by leveraging the W3C Web Annotation Ontology[3]. It includes all the needed abstractions to define annotations on the content of a target page. Another W3C recommendation in the same scope enables interoperation between clients and servers, i.e. the Web Annotation Protocol[4]. Although, this set of recommendations are not designed to serve the semantic annotations paradigm, it is a great source that could be leveraged to distill informa-

[2] W3C Web Annotation Vocabulary: https://www.w3.org/TR/annotation-vocab/

[3] W3C Web Annotation Ontology: https://www.w3.org/ns/oa

[4] W3C Web Annotation Protocol: https://www.w3.org/TR/annotation-protocol/

tion and interpret the content of the webpage by cosuming the metadata that the annotations encapsulate.

9.4 Envisioning the next Web era

As of August 2016, the most trending topics in the Web science include deep learning and personal assistants (also known as bots) based on Artifical Intelligence (AI). Internet of Things (IoT) is also on the hype with efforts from the respective industry leaders to interconnect the physical world with the online one in an attempt to provide added value to the corresponding user experience. However, the most fascinating trend and closely related to the presented research, is the development of personal assistants that are eager to enable the user to accomplish more tasks in a more efficient and effective way. Facebook announced at the second quarter of 2016 the support of bots in the messenger platform[5], which allows businesses to build bots that interact with the users in order to complete an action. Amazon introduced *Alexa*[6], the smart speaker that can be asked questions and play the role of personal assistant by consuming Web Services that have the ability to fulfil the requests of the user (e.g. retrieve the weather forecast, search for flights, etc.). In a similar fashion Google is developing the Google Home[7], which is playing the role of the personal assistant and also being able to forward any result to the mobile phone of the user (like directions on the map).

In this context, any website that is machine understandable can be considered eligible for consumption by the next generation of bots based on AI. In brief, a website need to provide explicit meaning to the presented information, declarations about the actions that can be completed and the way that an agent could interact with the webpage. Therefore, facilitating the transformation of websites to machine understandable websites could be considered equivalent to the implementation of a

[5] Facebook messenger bots announcement: http://newsroom.fb.com/news/2016/04/messenger-platform-at-f8/

[6] Amazon Alexa description: http://www.bloomberg.com/news/articles/2014-11-06/amazon-echo-is-a-listening-talking-music-playing-speaker-for-your-home

[7] https://googleblog.blogspot.gr/2016/05/io-building-next-evolution-of-google.html

separate API that would serve the website information in a structured way.

Connecting the aforementioned points, we could imagine the next Web era to build on top of a seamless integration of websites content with agents consuming structured data, mainly because the websites will provide structured data in the form of annotations on the presented content. In this landscape, the bots will be able to better serve the users with answers based on the humongous Web dataset and not to be limited in the consumption of a few specific predefined Web Services. The benefits of this setup would be multifold and for all the stakeholders, i.e. the user, the business provider, and the search engine. The user will get the answer that is seeking in a more comprehensive way, the business provider will reach the prospective consumers more efficiently by having more chances to be a candidate for a response to a user query, and finally the search engine will be able to better serve any endpoint that requests results to a given keyword. Even in the case of a website that the information is not related to a business relationship or calling for a transactional action, the search engine will be able to understand the content and draw insights out of it in order to compile a response for the given user request.

This unsupervised questions answering saves a lot of time for the user and if we consider the amount of transactions that can be completed with the ease of simply calling for them in natural language to a service endpoint like *Alexa*, we can realise how faster future users will complete simple tasks with decisions that can be easily declared in simple rules that the personal assistant will apply and make in an unsupervised way. Having more than one propositions from the industry as of the time of writing, sounds really promising for the realisation of the agents vision in the near future.

What is next if the agents vision sounds like an already old one that meets reality? In brief, I would say the answer refers to two directions: a) the leverage of the majority of the websites to machine understandable entities, therefore enabling the consumption of their content and actions by agents; and b) the leverage of the physical world entities to discoverable Web entities that can be connected and interact with agents. The former one is what the presented approach aims to facilitate, while the latter one is in the target of the Internet of Things (IoT) sector.

Appendix A

Manual Semantic Annotations

The survey assignment sheet and result stats

The purpose of Appendix A is to provide a detailed presentation of the task that was realised as both evaluation and survey in the scope of the PhD thesis. The detailed analysis of the data that was produced and the insights that were made are presented in Section 3.3, which presents the task from the survey perspective of it. Furthermore, the framework was compared to the answers of the evaluators, as described in Section 8.2, in order to measure the effectiveness of the automation that the developed approach introduces to the generation process of semantic annotations.

In this task, the participants were given the following set of questions and steps to complete. Main goal of this evaluation process is to get manually selected vocabulary terms sets and JSON-LD semantic annotations by evaluators familiar to Computer Science topics. Each participant was assigned to one of the four use cases below:

- *Article*: http://www.nasa.gov/feature/jpl/
  nasas-curiosity-rover-team-confirms-ancient-lakes-on-mars
- *Hotel*: http://www.mohr-life-resort.at/
  zimmer-und-preise/detail.html?rid=12
- *Museum*: http://www.louvre.fr/en/expositions/
  winged-victory-samothracerediscovering-masterpiece

- *Recipe*: http://www.cookingchanneltv.com/recipes/debi-mazar-and-gabriele-corcos/margherita-pizza.print.html

Going through the set of questions that were given to the evaluators, we see that the first two are related to the expertise of the participant regarding the semantic annotations topic, while the rest of the questionary is related to the generation of annotations for the specific use case they were given. Also, they were asked to time themselves for the questions 3 and 4, which are related to the generation of the annotations.

1. *How familiar are you with the topic of Semantic Annotations?* The lowest value would mean that it's your first time reading something related, while the highest, that you are an expert in the field. Range: [No idea - 0, Expert - 5]. Your expertise:

2. *If you are familiar with the Semantic Annotations, have you ever generated some?*
   ○ Yes
   ○ No

3. *Give a list of vocabulary terms that are suitable for annotating the given use case, by materialising the LOV search results. Which other candidates did you find? Why did you choose the term t and not another one?.*

4. *Based on the list above, construct the JSON-LD snippet that could be used together with the webpage of the examined use case.*
   The JSON-LD snippet will contain the vocabulary terms accompanied with the correspondant values. Hints: a) Check the property range; b) use a validator[1] to be sure that the produced JSON-LD has valid syntax.

5. *Which new terms would you propose for content that you wanted to annotate, but you couldn't find the appropriate existing vocabulary terms?*

[1] JSON-LD online validator: http://json-ld.org/playground/

Use Case	Expertise level						Vocabulary average time	JSON-LD average time
	0	1	2	3	4	5		
hotel	11	5	1	0	0	0	58.05	57.58
museum	14	5	0	0	0	0	41.57	41.21
article	10	4	0	1	0	0	60.20	48.10
recipe	9	3	1	0	0	0	40.15	42.15

Table A.1: Summary of the collected answers. The last two columns refer to the average time required by the participants to finish the tasks of vocabulary selection and JSON-LD generation respectively.

6. *Which was the most difficult step and why?*

Additionally, the evaluators were asked to record the time they needed to complete the steps 3 and 4. A summary of the raw survey data is presented in Table A.1.

The majority of the participants did not have priori experience to semantic annotations implementation based based on the answers of question 2, although some of them had heard or studied about the semantic annotations topic in the past.

For the sake of simplicity and better metrics application across the collected term URIs, the URIs have been normalised in a post-collection processing. Thus, all the URIs start with the http:// scheme and have been trasformed to lowercase tokens in order to be easier to group them with string similarity and provide the summarised metrics in Table A.2, Table A.3, Table A.4 and Table A.5.

Term URI	**Occurences**
http://schema.org/article	8
http://schema.org/author	7
http://schema.org/person	5
http://lsdis.cs.uga.edu/projects/semdis/opus#author	3
http://guava.iis.sinica.edu.tw/r4r/article	3
http://ns.nature.com/terms/person	3
http://purl.org/ontology/bibo/article	3
http://voag.linkedmodel.org/voag/#image	2
http://linguistics-ontology.org/gold/2010/article	2
http://schema.org/headline	2
http://sw-portal.deri.org/ontologies/swportal#article	2
http://dbpedia.org/ontology/image	2
http://spi-fm.uca.es/spdef/models/generictools/wikim/1.0#article	2
http://d-nb.info/standards/elementset/gnd#author	2
http://schema.org/newsarticle	2
http://lsdis.cs.uga.edu/projects/semdis/opus#article	2
http://schema.org/image	2
http://www.lexinfo.net/ontology/2.0/lexinfo#article	2
http://schema.org/datepublished	2
http://xmlns.com/foaf/0.1/person	2

Table A.2: Top 20 proposed vocabulary terms for the article use case. A long tail of 65 terms with solely one occurences follows the presented terms.

Term URI	**Occurences**
http://schema.org/hotel	7
http://dbpedia.org/ontology/hotel	4
http://purl.org/acco/ns#hotel	4
http://linkedgeodata.org/page/ontology/guesthouse	3
http://purl.org/acco/ns#hotelroom	3
http://schema.org/lodgingreservation	3
http://www.w3.org/2003/12/exif/ns#geo	2
http://linkedgeodata.org/ontology/hotel	2
http://dbpedia.org/ontology/skiresort	2
http://elite.polito.it/ontologies/dogont.owl#room	2
http://www.aktors.org/ontology/portal#has-web-address	2
http://schema.org/flightreservation	2
http://purl.org/ontology/bibo/image	2
http://purl.org/healthcarevocab/v1#tag.0010.1040	2
http://schema.org/skiresort	2
http://rdfs.co/juso/address	2
http://schema.org/sport	2
http://schema.org/trainreservation	2
http://linkedgeodata.org/page/ontology/hostel	2
http://schema.org/email	2

Table A.3: Top 20 proposed vocabulary terms for the hotel use case. A long tail of 72 terms with solely one occurences follows the presented terms, while 27 more terms occur twice.

Term URI	**Occurences**
http://schema.org/exhibitionevent	9
http://schema.org/museum	7
http://lod.nl.go.kr/ontology/exhibition	6
http://dbpedia.org/ontology/museum	5
http://schema.org/event	4
http://linkedgeodata.org/page/ontology/museum	4
http://purl.org/ontology/storyline/event	4
http://iflastandards.info/ns/fr/frbr/frbrer/p3046	4
http://vivoweb.org/ontology/core#museum	4
http://dbpedia.org/ontology/artpatron	4
http://www.aktors.org/#person	3
http://www.bbc.co.uk/ontologies/coreconcepts#terms_person	3
http://ontotext.com/proton-ontology/#art	3
http://xmlns.com/foaf/spec/	3
http://schema.org/person	2
http://metadataregistry.org/schemaprop/show/id/1728.html	2
http://linkedgeodata.org/page/ontology/artwork	2
http://sw-portal.deri.org/ontologies/swportal#eventnsl:event	2
http://sw-portal.deri.org/ontologies/swportal#haslocation	2
http://www.ontotext.com/proton/protonext#art	2

Table A.4: Top 20 proposed vocabulary terms for the museum use case. A long tail of 33 terms with solely one occurences follows the presented terms, while 6 more terms occur twice.

Term URI	#
http://data.lirmm.fr/ontologies/food#recipe	10
http://schema.org/recipe	8
http://schema.org/ingredients	7
http://semanticscience.org/resource/sio_001042	5
http://data.lirmm.fr/ontologies/food#ingredientlistastext	4
http://dbpedia.org/ontology/ingredient	4
http://schema.org/cookingmethod	3
http://data.lirmm.fr/ontologies/food#percent	3
http://data.lirmm.fr/ontologies/food#ingredient	3
http://kmi.open.ac.uk/projects/smartproducts/ontologies/food.owl#recipe	3
http://schema.org/recipeinstructions	2
http://schema.org/totaltime	2
http://semanticscience.org/resource/sio_001042.rdf	2
http://schema.org/cooktime	2
http://schema.org/recipeingredient	2
http://rdfs.co/bevon/ingredient	2
http://sensormeasurement.appspot.com/ont/home/homeactivity#cooking	2
http://schema.org/preptime	2
http://ogp.me/ns#image	1
http://www.w3.org/2001/sw/hcls/ns/transmed/tmo_0003	1

Table A.5: Top 20 proposed vocabulary terms for the recipe use case. A long tail of 46 terms with solely one occurences follows the presented terms.

Appendix B

Automatic Semantic Annotations

The list of examined webpages

The purpose of this appendix is to provide the list of webpaces that were used during the machine-based evaluation that was conducted without any intervention by the user. Table B.1 presents the list of webpages accompanied by their domain.

Domain	**URL**
Recipe	`http://www.cookingchanneltv.com/recipes/bbq-chicken-pizza`
Recipe	`http://allrecipes.com/recipe/15022/veggie-pizza/`
Recipe	`http://www.bbc.co.uk/food/recipes/flower_power_pizza_13681`
Recipe	`http://www.yummly.co/#recipe/Pizza-1106814`
Article	`http://www.bbc.com/news/science-environment-38920199`
Article	`http://www.independent.co.uk/news/science/nasa-announcement-press-conference-today-solar-system-exoplanet-sun-planets-news-latest-a7590281.html`
Local Business	`http://www3.hilton.com/en/hotels/berlin/hilton-berlin-BERHITW/index.html`
Local Business	`https://www.opentable.de/neni`

Table B.1: The URLs of the webpages used in the machine-based evaluation.

Bibliography

[1] Andrejs Abele, John P. McCrae, Paul Buitelaar, Anja Jentzsch, and Richard Cyganiak. Linking Open Data cloud diagram 2017. `http://lod-cloud.net/`, 2017.

[2] P Archer, S Goedertier, and N Loutas. Study on persistent URIs, with identification of best practices and recommendations on the topic for the MSs and the EC. *Interoperability Solutions for European Public Administrations*, 2012.

[3] Ghislain Auguste Atemezing and Raphaël Troncy. Information content based ranking metric for linked open vocabularies. In *Proceedings of the 10th International Conference on Semantic Systems*, pages 53–56. ACM, 2014.

[4] Sören Auer, Jan Demter, Michael Martin, and Jens Lehmann. LODStats–an extensible framework for high-performance dataset analytics. In *Knowledge Engineering and Knowledge Management*, pages 353–362. Springer, 2012.

[5] Cosmin Basca, Stéphane Corlosquet, Richard Cyganiak, Sergio Fernández, and Thomas Schandl. Neologism: Easy vocabulary publishing. 2008.

[6] Tim Berners-Lee. Cool {URIs} don\'t change. `https://www.w3.org/Provider/Style/URI.html`, 1998.

[7] Tim Berners-Lee. Linked data. `http://www.w3.org/DesignIssues/LinkedData.html`, 2009.

[8] Tim Berners-Lee, Roy Fielding, and Larry Masinter. Uniform resource identifier (URI): Generic syntax. Technical report, 2004.

[9] Nikos Bikakis, Chrisa Tsinaraki, Ioannis Stavrakantonakis, Nektarios Gioldasis, and Stavros Christodoulakis. The SPARQL2XQuery interoperability framework. *World Wide Web*, pages 1–88, 2014.

[10] Christian Bizer, Kai Eckert, Robert Meusel, Hannes Mühleisen, Michael Schuhmacher, and Johanna Völker. Deployment of RDFa, microdata, and microformats on the Web – a quantitative analysis. In *The Semantic Web–ISWC 2013*, pages 17–32. Springer, 2013.

[11] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.

[12] Dan Brickley. Announcing schema.org actions. `http://blog.schema.org/2014/04/announcing-schemaorg-actions.html`, 2014.

[13] Anila Sahar Butt. Ontology search: Finding the right ontologies on the Web. In *Proceedings of the 24th International Conference on World Wide Web Companion*, pages 487–491. International World Wide Web Conferences Steering Committee, 2015.

[14] Anila Sahar Butt, Armin Haller, and Lexing Xie. Relationship-based top-k concept retrieval for ontology search. In *Knowledge Engineering and Knowledge Management*, pages 485–502. Springer, 2014.

[15] Gavin Carothers and Eric Prud'hommeaux. RDF 1.1 Turtle. W3C recommendation, W3C, February 2014. https://www.w3.org/TR/turtle/.

[16] Philipp Cimiano, Siegfried Handschuh, and Steffen Staab. Towards the self-annotating Web. In *Proceedings of the 13th international conference on World Wide Web*, pages 462–471. ACM, 2004.

[17] Philipp Cimiano, Günter Ladwig, and Steffen Staab. Gimme'the context: context-driven automatic semantic annotation with C-PANKOW. In *Proceedings of the 14th international conference on World Wide Web*, pages 332–341. ACM, 2005.

[18] Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. GATE: an architecture for development of robust HLT applications. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 168–175. Association for Computational Linguistics, 2002.

[19] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *Proceedings of the 9th International Conference on Semantic Systems (I-Semantics)*, 2013.

[20] John Davies, York Sure, Denny Vrandecic, Sofia Pinto, Christoph Tempich, and York Sure. The DILIGENT knowledge processes. *Journal of Knowledge Management*, 9(5):85–96, 2005.

[21] Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, R. Guha, Anant Jhingran, Tapas Kanungo, Kevin S. McCurley, Sridhar Rajagopalan, Andrew Tomkins, John A. Tomlin, and Jason Y. Zien. A case for automated large-scale semantic annotation. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(1):115 – 132, 2003.

[22] Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, R Guha, Anant Jhingran, Tapas Kanungo, Sridhar Rajagopalan, Andrew Tomkins, John A Tomlin, et al. SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. In *Proceedings of the 12th international conference on World Wide Web*, pages 178–186. ACM, 2003.

[23] Ying Ding and Dieter Fensel. Ontology Library Systems: The key to successful ontology reuse. In *SWWS*, pages 93–112, 2001.

[24] John Domingue, Dieter Fensel, and James A Hendler. *Handbook of semantic web technologies*. Springer Science & Business Media, 2011.

[25] Mohamed Ben Ellefi, Zohra Bellahsene, Stefan Dietze, and Konstantin Todorov. Beyond established knowledge graphs-recommending Web datasets for data linking. In *European Conference on Web Engineering*, pages 262–279. Springer, 2016.

[26] Michael Erdmann, Alexander Maedche, H-P Schnurr, and Steffen Staab. From manual to semi-automatic semantic annotation: About ontology-based text annotation tools. In *Proceedings of the COLING-2000 Workshop on Semantic Annotation and Intelligent Content*, pages 79–85. Association for Computational Linguistics, 2000.

[27] Anna Fensel, Ioan Toma, José María García, Ioannis Stavrakantonakis, and Dieter Fensel. Enabling customers engagement and collaboration for small and medium-sized enterprises in ubiquitous multi-channel ecosystems. *Computers in Industry*, 65(5):891–904, 2014.

[28] Dieter Fensel. *Spinning the Semantic Web: bringing the World Wide Web to its full potential*. Mit Press, 2005.

[29] Roy Thomas Fielding. *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, Irvine, 2000.

[30] Giorgos Giannopoulos, Nikos Bikakis, Theodore Dalamagas, and Timos Sellis. GoNTogle: a tool for semantic annotation and search. In *Extended Semantic Web Conference*, pages 376–380. Springer, 2010.

[31] László Gönczy, András Kövi, András Pataricza, Andreas Thalhammer, Ioannis Stavrakantonakis, Thomas Cane, Audun Vennesland, Charalampos Doukas, and Maria Lambrou. D2.3 – The e-Freight Ontology. Technical report, European e-Freight capabilities for Co-modal transport, 05 2012.

[32] Google. Introducing schema.org: Search engines come together for a richer Web. `http://googlewebmastercentral.blogspot.co.at/2011/06/introducing-schemaorg-search-engines.html`, 2011.

[33] Thomas R Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993.

[34] Lavdim Halilaj, Irlán Grangel-González, Gökhan Coskun, Steffen Lohmann, and Sören Auer. Git4Voc: collaborative vocabulary development based on Git. *International Journal of Semantic Computing*, 10(02):167–191, 2016.

[35] Lavdim Halilaj, Niklas Petersen, Irlán Grangel-González, Christoph Lange, Sören Auer, Gökhan Coskun, and Steffen Lohmann. Vocol: an integrated environment to support version-controlled vocabulary development. In *Knowledge Engineering and Knowledge Management: 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings 20*, pages 303–319. Springer, 2016.

[36] Siegfried Handschuh and Steffen Staab. Authoring and annotation of Web pages in CREAM. In *Proceedings of the 11th International Conference on World Wide Web*, WWW '02, pages 462–473, New York, NY, USA, 2002. ACM.

[37] Andreas Harth. Billion Triples Challenge data set. Downloaded from http://km.aifb.kit.edu/projects/btc-2012/, 2012.

[38] Tom Heath and Christian Bizer. Linked Data: Evolving the Web into a global data space. *Synthesis lectures on the semantic web: theory and technology*, 1(1):1–136, 2011.

[39] James Hendler. Is there an intelligent agent in your future? *Nature*, 11, 1999.

[40] Martin Hepp. Goodrelations: An ontology for describing products and services offers on the Web. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 329–346. Springer, 2008.

[41] Annika Hinze, Ralf Heese, Markus Luczak-Rösch, and Adrian Paschke. Semantic enrichment by non-experts: usability of manual annotation tools. In *The Semantic Web–ISWC 2012*, pages 165–181. Springer, 2012.

[42] Aidan Hogan, Andreas Harth, Alexandre Passant, Stefan Decker, and Axel Polleres. Weaving the pedantic Web. *3rd International Workshop on Linked Data on the Web (LDOW2010)*, 2010.

[43] Tobias Käfer and Andreas Harth. Billion Triples Challenge data set. Downloaded from http://km.aifb.kit.edu/projects/btc-2014/, 2014.

[44] Ali Khalili and Sören Auer. User interfaces for Semantic authoring of textual content: A systematic literature review. 2012.

[45] Ali Khalili and Sören Auer. WYSIWYM authoring of structured content based on schema.org. In *Web Information Systems Engineering–WISE 2013*, pages 425–438. Springer, 2013.

[46] Graham Klyne. Uniform resource identifier (URI) schemes. `http://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml`, 2015.

[47] Paul A Kogut and William S Holmes III. AeroDAML: Applying information extraction to generate DAML annotations from Web pages. In *Semannot@ K-CAP 2001*, 2001.

[48] Michal Laclavik, Martin Seleng, Emil Gatial, Zoltan Balogh, and Ladislav Hluchy. Ontology based text annotation-OnTeA. *Frontiers in Artificial Intelligence and Applications*, 154:311, 2007.

[49] Michal Laclavík, Martin Šeleng, and Ladislav Hluchỳ. Towards large scale semantic annotation built on mapreduce architecture. In *International Conference on Computational Science*, pages 331–338. Springer, 2008.

[50] Steve Lawrence, David M Pennock, Gary William Flake, Robert Krovetz, Frans M Coetzee, Eric Glover, Finn Årup Nielsen, Andries Kruger, and C Lee Giles. Persistence of web references in scientific research. *Computer*, (2):26–31, 2001.

[51] Alexander Linden and Jackie Fenn. Understanding gartner's hype cycles. *Strategic Analysis Report N° R-20-1971. Gartner, Inc*, 2003.

[52] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[53] David Marcus. Testing phase announcement of the Facebook digital assistant M. `https://www.facebook.com/Davemarcus/posts/10156070660595195`, 2015.

[54] Diana Maynard, Valentin Tablan, Kalina Bontcheva, Hamish Cunningham, and Yorick Wilks. Muse: a multisource entity recognition system. *Computers and the Humanities. Website Reference: http://gate. ac. uk/sale/muse/muse. pdf*, 2003.

[55] Robert Meusel and Heiko Paulheim. Heuristics for fixing common errors in deployed schema.org microdata. In *The Semantic Web. Latest Advances and New Domains–ESWC 2015*, pages 152–168. Springer, 2015.

[56] Robert Meusel, Petar Petrovski, and Christian Bizer. The WebDataCommons Microdata, RDFa and microformat dataset series. In *The Semantic Web–ISWC 2014*, pages 277–292. Springer, 2014.

[57] Yassine Mrabet, Claire Gardent, Muriel Foulonneau, Elena Simperl, and Eric Ras. Towards knowledge-driven annotation. In *AAAI*, pages 2425–2431, 2015.

[58] Roberto Navigli and Simone Paolo Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

[59] Günter Neumann, Rolf Backofen, Judith Baur, Markus Becker, and Christian Braun. An information extraction core system for real world german text processing. In *Proceedings of the fifth conference on Applied natural language processing*, pages 209–216. Association for Computational Linguistics, 1997.

[60] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The PageRank citation ranking: Bringing order to the Web. Technical Report 1999-66, Stanford InfoLab, November 1999.

[61] Peter F Patel-Schneider. Analyzing schema.org. In *The Semantic Web–ISWC 2014*, pages 261–276. Springer, 2014.

[62] Ken Peffers, Tuure Tuunanen, Marcus A Rothenberger, and Samir Chatterjee. A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77, 2007.

[63] Borislav Popov, Atanas Kiryakov, Angel Kirilov, Dimitar Manov, Damyan Ognyanoff, and Miroslav Goranov. KIM–semantic annotation platform. In *International Semantic Web Conference*, pages 834–849. Springer, 2003.

[64] Lawrence Reeve and Hyoil Han. Survey of semantic annotation platforms. In *Proceedings of the 2005 ACM symposium on Applied computing*, pages 1634–1638. ACM, 2005.

[65] Ellen Riloff and Jessica Shepherd. A corpus-based approach for building semantic lexicons. *arXiv preprint cmp-lg/9706013*, 1997.

[66] Antonio Roa-Valverde, Andreas Thalhammer, Ioan Toma, and Miguel-Angel Sicilia. Towards a formal model for sharing and reusing ranking computations. In *Proc. of the 6th Intl. Workshop on Ranking in Databases In conjunction with VLDB*, volume 2012, 2012.

[67] Ian Rogers. The Google pagerank algorithm and how it works, 2002.

[68] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295. ACM, 2001.

[69] Johann Schaible, Thomas Gottron, Stefan Scheglmann, and Ansgar Scherp. LOVER: support for modeling data using Linked Open Vocabularies. In *Proceedings of the Joint EDBT/ICDT 2013 Workshops*, pages 89–92. ACM, 2013.

[70] Johann Schaible, Thomas Gottron, and Ansgar Scherp. Survey on common strategies of vocabulary reuse in Linked Open Data modeling. In *The Semantic Web: Trends and Challenges*, pages 457–472. Springer, 2014.

[71] Johann Schaible, Thomas Gottron, and Ansgar Scherp. TermPicker: Enabling the reuse of vocabulary terms by exploiting data from the Linked Open Data cloud. In *European Semantic Web Conference*, pages 101–117. Springer, 2016.

[72] Thomas Schandl and Andreas Blumauer. Poolparty: SKOS thesaurus management utilizing linked data. In *Extended Semantic Web Conference*, pages 421–425. Springer, 2010.

[73] Max Schmachtenberg, Christian Bizer, Anja Jentzsch, and Richard Cyganiak. Linking open data cloud diagram 2014. `http://lod-cloud.net`, 2014.

[74] Elena Simperl. Reusing ontologies on the Semantic Web: A feasibility study. *Data & Knowledge Engineering*, 68(10):905–925, 2009.

[75] Elena Simperl and Markus Luczak-Rösch. Collaborative ontology engineering: a survey. *The Knowledge Engineering Review*, 29(01):101–131, 2014.

[76] Steffen Stadtmüller, Andreas Harth, and Marko Grobelnik. Accessing information about linked data vocabularies with vocab.cc. In *Semantic Web and Web Science*, pages 391–396. Springer, 2013.

[77] Ioannis Stavrakantonakis. Personal data and user modelling in tourism. In *Information and Communication Technologies in Tourism 2013*, pages 507–518. Springer, 2013.

[78] Ioannis Stavrakantonakis, Anna Fensel, and Dieter Fensel. Matching Web entities with potential actions. In *SEMANTiCS 2014*, 2014.

[79] Ioannis Stavrakantonakis, Anna Fensel, and Dieter Fensel. Linked Open Vocabulary recommendation based on ranking and linked open data. In *Joint International Semantic Technology Conference*, pages 40–55. Springer International Publishing, 2015.

[80] Ioannis Stavrakantonakis, Anna Fensel, and Dieter Fensel. Linked Open Vocabulary ranking and terms discovery. In *Proceedings of the SEMANTiCS 2016*. ACM, 2016.

[81] Ioannis Stavrakantonakis, Anna Fensel, and Dieter Fensel. Towards a vocabulary terms discovery assistant. In *SEMANTiCS 2016*, 2016.

[82] Ioannis Stavrakantonakis, Andreea-Elena Gagiu, Harriet Kasper, Ioan Toma, and Andreas Thalhammer. An approach for evaluation of social media monitoring tools. *Common Value Management*, 52, 2012.

[83] Ioannis Stavrakantonakis, Andreas Thalhammer, Alex Oberhauser, Corneliu-Valentin Stanciu, and Ioan Toma. D2.4/D2.5 – e-Freight Semantic Registry and Repository / e-Freight SESA platform. Technical report, European e-Freight capabilities for Co-modal transport, 04 2013.

[84] Ioannis Stavrakantonakis, Andreas Thalhammer, Alex Oberhauser, Corneliu-Valentin Stanciu, Ioan Toma, Audun Vennesland, and Thomas Cane. Introduction of the Semantically Enabled Service Architecture to the freight domain. *2nd International Conference on Applied Paperless Freight Transport and Logistics*, 2012.

[85] Ioannis Stavrakantonakis, Ioan Toma, Anna Fensel, and Dieter Fensel. Hotel websites, Web 2.0, Web 3.0 and online direct marketing: The case of Austria. In *Information and Communication Technologies in Tourism 2014*, pages 665–677. Springer International Publishing, 2014.

[86] Thomas Steiner, Raphael Troncy, and Michael Hausenblas. How Google is using linked data today and vision for tomorrow. *Linked Data in the Future Internet at the Future Internet Assembly (FIA 2010), Ghent*, 2010.

[87] Armando Stellato, Sachit Rajbhandari, Andrea Turbati, Manuel Fiorelli, Caterina Caracciolo, Tiziano Lorenzetti, Johannes Keizer, and Maria Teresa Pazienza. VocBench: a web application for collaborative development of multilingual thesauri. In *European Semantic Web Conference*, pages 38–53. Springer, 2015.

[88] Andreas Thalhammer, Ioannis Stavrakantonakis, and Ioan Toma. Diversity-aware clustering of SIOC posts. In *I-SEMANTICS (Posters & Demos) 2013*. Citeseer, 2013.

[89] Ioan Toma, Corneliu Stanciu, Anna Fensel, Ioannis Stavrakantonakis, and Dieter Fensel. Improving the online visibility of touristic service providers by using semantic annotations. In *The Semantic Web: ESWC 2014 Satellite Events*, pages 259–262. Springer International Publishing, 2014.

[90] Pierre-Yves Vandenbussche, Ghislain A Atemezing, María Poveda-Villalón, and Bernard Vatant. Linked Open Vocabularies (LOV): a gateway to reusable semantic vocabularies on the Web. *Semantic Web*, (Preprint):1–16, 2015.

[91] Pierre-Yves Vandenbussche and Bernard Vatant. Linked Open Vocabularies. *ERCIM news*, 96:21–22, 2014.

[92] Maria Vargas-Vera, Enrico Motta, John Domingue, Mattia Lanzoni, Arthur Stutt, and Fabio Ciravegna. MnM: Ontology driven semi-automatic and automatic support for semantic markup. In *International Conference on Knowledge Engineering and Knowledge Management*, pages 379–391. Springer, 2002.

[93] Valentin Zacharias and Simone Braun. SOBOLEO–social bookmarking and lighweight engineering of ontologies. *CKC*, 273, 2007.