Web-Scale Profiling of Semantic Annotations in HTML Pages

Inauguraldissertation
zur Erlangung des akademischen Grades
eines Doktors der Naturwissenschaften
der Universität Mannheim

vorgelegt von
Dipl.-Wirtsch.-Inf. Robert Meusel
aus Berlin

Mannheim, 2016

Dekan: Professor Dr. Heinz Jürgen Müller, Universität Mannheim
Referent: Professor Dr. Christian Bizer, Universität Mannheim
Korreferent: Professor Dr. Wolfgang Nejdl, Leibniz Universität Hannover

Tag der mündlichen Prüfung: 10. März 2017

Abstract

The vision of the *Semantic Web* was coined by Tim Berners-Lee almost two decades ago. The idea describes an extension of the existing Web in which "information is given well-defined meaning, better enabling computers and people to work in cooperation" [Berners-Lee et al., 2001].

Semantic annotations in HTML pages are one realization of this vision which was adopted by large numbers of web sites in the last years. Semantic annotations are integrated into the code of HTML pages using one of the three markup languages Microformats, RDFa, or Microdata. Major consumers of semantic annotations are the search engine companies Bing, Google, Yahoo!, and Yandex. They use semantic annotations from crawled web pages to enrich the presentation of search results and to complement their knowledge bases.

However, outside the large search engine companies, little is known about the deployment of semantic annotations: How many web sites deploy semantic annotations? What are the topics covered by semantic annotations? How detailed are the annotations? Do web sites use semantic annotations correctly? Are semantic annotations useful for others than the search engine companies? And how can semantic annotations be gathered from the Web in that case?

The thesis answers these questions by profiling the web-wide deployment of semantic annotations.

The topic is approached in three consecutive steps: In the first step, two approaches for extracting semantic annotations from the Web are discussed. The thesis evaluates first the technique of focused crawling for harvesting semantic annotations. Afterward, a framework to extract semantic annotations from existing web crawl corpora is described. The two extraction approaches are then compared for the purpose of analyzing the deployment of semantic annotations in the Web.

In the second step, the thesis analyzes the overall and markup language-specific adoption of semantic annotations. This empirical investigation is based on the largest web corpus that is available to the public. Further, the topics covered by deployed semantic annotations and their evolution over time are analyzed. Subsequent studies examine common errors within semantic annotations. In addition, the thesis analyzes the data overlap of the entities that are described by semantic annotations from the same and across different web sites.

The third step narrows the focus of the analysis towards use case-specific issues. Based on the requirements of a marketplace, a news aggregator, and a travel portal the thesis empirically examines the utility of semantic annotations for these use cases. Additional experiments analyze the capability of product-related semantic annotations to be integrated into an existing product categorization schema. Especially, the potential of exploiting the diverse category information given by the web sites providing semantic annotations is evaluated.

Zusammenfassung

Vor mehr als 20 Jahren veröffentlichte Tim Berners-Lee seine Idee des *Semantic Web*s. Basierend auf seiner Vision, sollte das semantische Web eine Erweiterung des bestehenden Webs sein, in dem die enthaltenen Informationen semantisch definiert sind, wodurch die Kooperation zwischen Mensch und Maschine vereinfacht werden würde. [Berners-Lee et al., 2001]

Semantische Annotationen in HTML Seiten sind eine konkrete Umsetzung dieser Idee, die in den letzten Jahren von sehr vielen Webseitenbetreibern adaptiert wurden. Semantische Annotationen werden direkt im HTML Quellcode der Webseite mithilfe der drei HTML-Markup-Erweiterungen Microformats, RDFa, und Microdata eingefügt. Hauptsächlich werden so annotierte Informationen von den großen Suchmaschinenfirmen, Bing, Google, Yahoo! oder Yandex verarbeitet. Diese Firmen nutzen semantische Annotationen, die sie in dem HTML Quellcode von gecrawlten Webseiten finden, um die Anzeige von Suchergebnissen zu verbessern oder ihren internen Wissensgraphen zu erweitern. Trotz der starken Nutzung durch die Suchmaschinenfirmen ist wenig über die Einbindung und Verbreitung von semantischen Annotationen im Web bekannt: Wie viele Webseiten bieten semantische Annotationen an? Welche Themengebiete werden beschrieben? Wie detailliert sind die annotierten Informationen und nutzen Webseitenbetreiber die Annotationen korrekt? Sind die so angebotenen Informationen nützlich und wie können sie effizient gesammelt werden?

Diese Fragen werden in den drei, aufeinanderfolgenden Teilen dieser Dissertation im Zuge einer umfassenden Profilierung des Datenraumes, der von semantischen Annotationen aufgespannt wird, beantwortet.

Im ersten Teil werden zwei Möglichkeiten zur Sammlung von semantischen Annotationen diskutiert. Zuerst evaluiert die Dissertation eine Methodik, die sich an der Idee des fokussierten Crawlens orientiert. Daraufhin wird ein Framework vorgestellt, welches semantische Annotationen aus bestehenden Webcrawldatensätzen extrahieren kann. Beide Vorgehensweisen werden verglichen und mit Bezug auf die Repräsentativität der gewonnen Daten evaluiert.

Im zweiten Teil analysiert die Arbeit empirisch die allgemeine, wie auch markupspezifische Verbreitung von semantischen Annotationen im Web basierend auf dem größten öffentlich zugänglichen Webcrawldatensatzes. Über die Verbreitung hinaus werden die enthaltenen Themengebiete sowie deren Veränderung über die Zeit betrachtet. Nachfolgend untersucht die Arbeit, zu welchem Grad Webseitenbetreiber semantische Annotationen korrekt benutzen.

Der abschließende Teil der Arbeit fokussiert sich auf eine anwendungsbezogene Analyse von semantischen Annotationen. Basierend auf den Anforderungen eines Onlineshops, einer Nachrichtenaggregationsseite und eines Reiseportals wird die Nützlichkeit von semantischen Annotationen evaluiert. Anschließend wird untersucht, in wie weit es möglich ist, die seitenspezifischen Produktkategorisierungen zu nutzen um Produktinformationen, auf eine bestehende Produktklassifizierung abzubilden und somit eine feingranulare Themenanalyse zu ermöglichen.

Contents

Chapter 1

Introduction

The vision of the *Semantic Web* was described by Tim Berners-Lee almost two decades ago [Berners-Lee et al., 2001]. He thought of the semantic web as an extension to the – at that point in time – current Web, offering humans and machines an improved communication. Machines should be empowered to understand the content, the interactions, and the transactions in the Web. Berners-Lee especially emphasized the necessity to infuse information with their well-defined meaning with in the documents of the Web, to realize his vision.

Besides other examples, this necessity of well-defined meaning becomes more clear with respect to the information contained in ordinary web pages. Humans can make sense of the content presented by a page, due to their background knowledge and the intend of requesting this particular page. Machines, parsing the underlying HTML code, might have difficulties in identifying the important content within the document and making sense of the presented information. Exemplary, it needs to be decisive for a machine that the term *Bremen* is the major concept of a web page and represents a city in Germany and not a city in the United States or a ship in the particular context.

In the last decade, a large body of research has been going on, focusing on various aspects of the realization of the idea of the semantic web. New and adapted approaches have been studied and evaluated to infuse a concrete meaning to information provided in the Web. Data formats like *RDF* have been developed providing a universal technique to exchange such information [Klyne and Carroll, 2004]. Specialized systems like *triple stores* and query languages like *SPARQL* have been created and refined to efficiently store and query such kind of information [Prud'Hommeaux et al., 2008]. Ontologies and vocabularies were extended and designed to cover (parts of) the topics described by information contained in the Web.

A concrete (partly) realization of the vision of the semantic web is called *linked open data* (LOD). The idea is to create and maintain collections of well-defined information (datasets), containing descriptions of various entities. If possible, the entities should be connected to related entities within the same dataset, as well as

across all LOD datasets. The resulting connected graph of datasets is referred to as the *LOD cloud*. The data, as the name indicates, should be open and publicly available, allowing consumers to access the data at any time. Based on the analysis of the LOD cloud by [Schmachtenberg et al., 2014], 1 014 datasets are contained, describing various topics such as *government*, *publications*, and *life sciences*. In comparison to the overall number of different data providers (e.g., web sites) in the Web as well the variety of covered topics, the number of datasets contained in the LOD cloud and their topical coverage seem rather limited.

Semantic Annotations Another, more recent, concrete realization of the vision of the semantic web is the inclusion of the meaning of information directly in the underlying code of the HTML page. The technical idea is based on the extension of the standard HTML markup by further semantic markup languages, which define additional sets of attributes and can be automatically recognized, e.g. by a machine. The most observed semantic markup languages are Microformats, RDFa, and Microdata. In order to infuse the embedded information with meaning, vocabularies are used to describe the resources and their attributes/properties.

One of the currently known, major consumers of semantic annotations are the large search engine companies Bing, Google, Yahoo!, and Yandex. Although not all services where semantic annotations are used for by those companies are known, one of the most popular ones is *Google's Rich Snippets*[1] [Goel et al., 2009]. For a selected set of topics (e.g., products and events), Google enriches the displayed search results with additional information retrieved from semantic annotations. As studies have shown, such enhanced presentations of search results are potentially more attractive to users [Cutrell and Guan, 2007]. The thereby generated improved visibility of data providers[2] leads to a win-win situation for data providers and data consumers.

Figure 1.1 depicts a web page of the online store of the sporting goods and fashion provider *Adidas*. A human visitor of this web page can directly identify that a shoe with the name *ace 16.3 indoor shoe* is offered for the price of US$70. Whenever we ask a machine to try to understand the content of this particular web page the results might vary. A machine might for example mix up the real price with the model number 16.3. In addition, depending on the underlying HTML code, there might be additional, partly unrelated information which might not even been shown by the browser at all (e.g., meta information, or information which are commented out).

[1] `https://developers.google.com/structured-data/rich-snippets/`

[2] Within this thesis we use the terms *data provider*, *web site*, *pay-level domain* (PLD) and *domain* to describe the same administrative authority, namely the person or group which is responsible for the content (data) provided within HTML pages belonging to one web site which are public available within the Web.

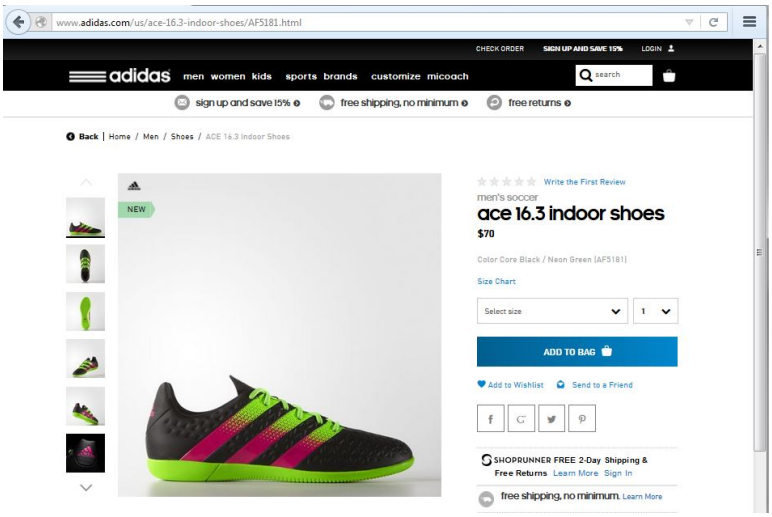


Figure 1.1: Example web page showing an Adidas soccer shoe.

Listing 1.1 shows an excerpt of the HTML code which is used in order to create parts of the human-visible web page displayed in Figure 1.1.[3]

Listing 1.1: HTML code excerpt of an Adidas shoe page.

```
<html > ...
<body>...
  <div>
  <span>men's soccer</span>
  <h1>ACE 16.3 Indoor Shoes</h1>
  <div>
    <meta content="USD">
    <span>70</span>
  </div>
  </div>...
</body>
</html>
```

As said before, although the reduced HTML code might be easier to understand by a machine it still is unclear, what kind of item is described. It can be a product, or an event and its just the entrance fee for a soccer court.

To overcome this issue, following the idea of the semantic web, semantic annotations enable the infusion with concrete meaning directly in the HTML code. Listing 1.2 shows an extended version of the former HTML code, where with the help of a small set of additional HTML attributes (in this example from the semantic markup language *Microdata*) and terms from a fictional vocabulary the described entity is semantically annotated.[4] The additional HTML attributes together with the

[3] For reasons of exemplification we have adapted the original used HTML code of this particular Adidas web page. We simplified the original HTML code by removing mainly css-class identifiers, JavaScript, and additional code used for tracking and the creation of dynamic elements.

[4] The presentation is simplified and the actual schema which is used by *Adidas* is removed due to reasons of understandability. At the current state (March 2016) Adidas makes use of schema.org.

attribute values from the vocabulary create the same visual presentation of the web page in the browser than the code of Listing 1.1.[5] In addition, the attributes and the vocabulary allow a machine to programmatically extract the information from the web page and understand them based on the definition of the vocabulary. From the code in Listing 1.2 the information can be extracted, that the web page presents something that is a `Product`. The product is `name`d *ACE 16.3 Indoor Shoes*. In addition, something which is an `Offer` is adhered to the product. This offer has a `price` of 70 of the `priceCurrency` *USD*.

Listing 1.2: HTML code excerpt of an annotated Adidas shoe page.

```
1  <html >...
2  <body>...
3    <div itemscope itemtype="Product">
4    <span>men's soccer</span>
5    <h1 itemprop="name">ACE 16.3 Indoor Shoes</h1>
6    <div itemprop="offers" itemscope itemtype="Offer">
7      <meta itemprop="priceCurrency" content="USD">
8      <span itemprop="price">70</span>
9    </div>
10   </div>...
11 </body>
12 </html>
```

Making use of such small extensions of the existing HTML code of a web page allows the programmatic extraction and understanding of the described information.

1.1 Motivation

The before mentioned, relatively simple example already reveals the major benefits of semantic annotations. As the semantic markup language is included directly in the HTML code, no additional technical resources (e.g., servers or databases) are necessary to provide the information to consumers. In addition, the syntax of semantic markup language follows the syntax of the HTML standard which further reduces the entry barrier for web site providers, as they are aware of the HTML markup language. As the inclusion of semantic markup languages do not change the layout of the web page the same document (HTML page) can be used to transport information to the human reader (visitor requesting the page via a web browser) as well as a machine, programmatic parsing the page.

As stated already before, the major consumer of semantic annotations are the search engine companies Bing, Google, Yahoo!, and Yandex. Based on the insights given in [Guha et al., 2015], they make use of knowledge extracted from semantic annotations for various application. Unfortunately not all of them are known, but some of them directly offer benefits for data providers.

As within the Web – to some extend – everybody can access each web page, semantic annotations can be exploited also by others than the search engine compa-

[5] The technical foundations of semantic annotations in HTML pages are described in Chapter 2 in detail.

nies. Furthermore, in contrast to other data sources like the LOD could, web pages are updated more frequently and hence semantic annotations in the HTML pages might be more up-to-date. In addition, we think that due to the general broad topical coverage of the Web, various application domains can potentially benefit from semantic annotations extracted from HTML pages. Existing services like product comparison platforms (e.g., *pricegrabber*[6]) can directly collect information about products from other web sites by parsing the semantic annotations embedded in the underlying HTML code. Currently, most of those services need to be connected manually to the different data sources, e.g., by implementing an API, importing data dumps, or writing wrapper for the desired web site. Other services which potentially can benefit from semantic annotations are recommendation service, who can collection additional information about recommended items, like books, movies and music. Those information can help to improve the quality and relatedness of suggested items to the user, as mentioned in [Ristoski et al., 2014].

Although the number of potential use cases for semantic annotations is huge, semantic annotations have not been considered too often, at least in the area of science.[7] Especially in the comparison to the former and ongoing research in the closely related research field focusing on LOD data, the number of works using semantic annotations is comparable low.

1.2 Problem Description and Contributions

We belief that the main reason for the missing adoption of semantic annotations in science and other applications domain is based on the lack of knowledge about semantic annotations in general.

So far, only the big search engine companies could gather large amounts of such data. Their core business consists partly of the indexing of the Web, thus they collect the content of a large amount of web pages anyway. From the collected HTML pages they can parse the semantic annotations and exploit them for various of their services. Unfortunately, they do not publish too many information about the spread, the topical coverage or the quality of semantic annotations. This is not surprising as the data is part of their business value. The publication of the data and/or statistics contradicts their business model.

In general, knowledge about a data is absolutely necessary before taking the exploitation of the data for a particular application into consideration. Therefore, the overall goal of this thesis is to overcome this lack of knowledge for the domain of semantic annotations. Making use of empirical studies, the thesis profiles different aspect of semantic annotations, answering questions about the general adoption, the topical coverage, the cleanness, as well as the use case-specific utility.

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

[7] We need to restrict us to the field of scientific research, as we do not know to which extend semantic annotations are considered in commercial applications, besides those of the search engine companies and Facebook.

The presented results and findings allow potential data consumers (researchers, commercial and non-commercial services) to judge the potential and limitations of semantic annotations in HTML pages. Furthermore, the insights should inspire further researchers to continue the exploration of semantic annotations and exploit them for different approaches. We approach the overall goal in three consecutive steps. The particular challenges of each of the three steps and the contributions provided by the thesis are described in the following.

Data Availability The first, major barrier, which needs to be overcome when trying to build a representative profile for semantic annotations in HTML pages is the collection large amounts of data. This task is challenging with respect to various aspects in terms of resources as well as knowledge of the overall structure of the Web, as described in [Rheinländer et al., 2016].

Until mid of 2012 only a limited number of organizations like the four big search engine companies were able to collect a larger amount of web pages. Hence the companies were able to profile the characteristics of the deployment and topical coverage of semantic annotations. In 2012, the *Common Crawl Foundation* (CC)[8] started to continuously release crawled web corpora of a decent size and made them publicly available. Each of the corpora contains several tera-bytes of compressed HTML pages. The availability of these web corpora enables alternative ways to collection web pages containing semantic annotations. At the same time, when working with such web crawl corpora, the sheer size raises another challenge, as in most cases only limited resources, time- and moneywise, are available. Still, it is not known which alternatives are practicable and how the chosen alternative influence the characteristics of the collected data.

Therefore, the thesis analyzes two different ways how semantic annotations can be gathered. First, an adapted focused crawling approach is evaluated based on its capability of efficiently crawling web pages making use of semantic markup languages. Further, the extraction from existing web corpora is evaluated, using a scalable cloud-based framework. In addition, as one of the first, we compare both approaches and comprehensively discuss the aspect of representativity. This is necessary, as both approaches cannot collect all semantic annotations contained in the Web, and therefore, all subsequent studies are based on a sample of the Web.

General Analysis Until today, we have only limited and unreliable knowledge about how many web sites make use of the semantic markup languages, what vocabularies are used and what topics are covered by semantic annotations. This lack of general insights makes it difficult for interested individuals or organizations to get an idea if semantic annotations can potentially be helpful for their applications and use cases. Furthermore, a common prejudice about real-world data is that the data is *dirty*, especially web data [Hernández and Stolfo, 1998]. Dirty data is in most cases not directly usable for any application. So far, it is also unknown to

[8] `http://commoncrawl.org`

which extend semantic annotations are affected by this prejudice, and with how much effort they can be made applicable.

We perform – as one of the first – an analysis of the overall deployment of semantic annotations in the Web and its evolution between 2012 and 2014. The conducted empirical studies are not limited to the general spread in the Web, but examine the topical coverage and distribution of semantic annotations. Furthermore, the dirtiness of semantic annotations is analyzed in terms of duplicates as well as compliance to the vocabulary definition. The collected insights are used to build a cleansing pipeline for semantic annotations which incorporates simple, but effective heuristics to overcome most compliance issues. In addition, we present an analysis measuring the impact of cleansing and duplicate removal on the topical distribution of semantic annotations. In a subsequent study, the thesis analyzes the interaction between vocabulary definition and the adoption in the Web over time. The findings reveal that changes are adopted quickly, but the adoption is mainly driven through incentives.

Use Case-Specific Analysis Besides the lack of knowledge about general aspects of semantic annotations like the overall deployment in the Web, also little is known about the concrete utility of semantic annotations for specific use cases. Meaning, even if an organization has identified that semantic annotations from some sources potentially fit to the general scope of their application, it is unknown if the use case-specific requirements (e.g., presents of certain attributes) can be fulfilled.

As example, a marketplace like *Ebay* wants to provide for each offered product additional information from other shops, as a description, competitive prizes, or ratings and reviews. Up till now, it is unclear if such information are contained in semantic annotations and how many web sites provide those information. In addition, even if the necessary information are available, so far no study has focused how they can be integrated in existing applications ore be combined to new services.

As the analyzes of the utility of semantic annotations cannot be performed in a generic fashion, the thesis carries out studies of the utility of semantic annotations based on the information requirements of three use cases, namely a *marketplace*, a news *aggregator*, and a *travel portal*. The findings reveal, that the utility is highly use case dependent, where especially for the third use case the provided data is insufficient. A subsequent study focuses on the examination of a more fine-grained topical distribution of product-related semantic annotations. The goal is to gain further knowledge about the product categories (e.g., electronics) which are covered by semantic annotations. Using a distant-supervised approach, which omits the necessity of manual crafted training data, the thesis uncovers potential flaws in the quality of this part of the data. The result underline the need for domain-specific profiling methods to gain further insights into the topics of semantic annotations.

1.3 Thesis Outline

In this section, the content of each following chapter is summarized and the major contributions are stated.

Chapter 2: Preliminaries Having introduced and motivated the goal of the thesis, this chapter presents the (technical) foundations of semantic annotations. In particular, the chapter presents the three different semantic markup languages and the most important vocabularies in the context of semantic annotations. In a last section, the chapter explains to which extend semantic annotations can be recognizes as own dataspace.

PART I: Extraction of Semantic Annotations

This part of the thesis describes two alternative strategies to harvest semantic annotations from the Web. First, the adaptability of focused crawling for this purpose is evaluated. Afterward, a framework is introduced allowing the scalable extraction of semantic annotations from public web corpora. The two approaches are evaluated based on their general potentials and drawbacks, as well as for the purpose of analyzing the adoption of semantic annotations in the Web. Especially the dimension of representativity is discussed.

Chapter 3: Data Extraction from the Web using Focused Crawling. Focused crawling describes the idea of steering a crawler to especially prefer and harvest web pages containing information which are useful for a specific task. This chapter evaluates to which extend focused crawling can be adopted to steer crawlers to collect preferably HTML pages containing semantic annotations. Based on a large-scale evaluation we show that by extending the state-of-the-art focused crawling approach with a bandit-based selection strategy, the harvesting rate is increased by further 25%.

Chapter 4: Data Extraction from Web Crawl Corpora. Besides the possibility of using focused crawling to collect semantic annotations the extraction from existing web corpora is possible. We present an approach which allows the scalable extraction of semantic annotations from such corpora, which recently become available to the public. The implemented approach makes use of the cloud-based, on-demand infrastructure AWS, and is evaluated by extracting semantic annotations from three different tera-byte-sized corpora. Its adaptability is shown by a number of related projects, making use of the general infrastructure of the framework.

Chapter 5: Comparison of the Extraction Approaches. Having evaluated two different alternatives for collect semantic annotations, this chapter discusses the influence of the selected harvesting strategy for later profiling and usage of the data. In particular the aspect of representativity with respect to the public Web is discussed. The section shows that semantic annotations extracted from existing web corpora are most suitable for the scope of this thesis. For those samples of the Web the problem of sampling errors is discussed, where we further show that the expected sampling error is sufficiently small.

PART II: Analysis of the Deployment of Semantic Annotations

The second part of the thesis focuses on the use case-independent analysis of semantic annotations. In particular, the overall adoption as well as the topical coverage of semantic annotations are analyzed. Furthermore, the quality of semantic annotations is inspected and methods are proposed to overcome data quality issues. Especially the influence on the resulting profile is shown. In a third step, the adoption of changes within the definition of the vocabulary is analyzed, which reveals potential insights about the future topical coverage and compliance.

Chapter 6: Overall Adoption of Semantic Annotations. This chapter introduces the different dimensions of profiling and discusses related work. In the following, first, the general deployment of semantic annotations in the Web within the last years is analyzed. Subsequent, the topical coverage and the markup language-specific deployment is studied. The findings of this chapter indicate a strong increasing adoption of semantic annotations in the Web in general, where especially Microdata is becoming more and more popular. Topical-wise, we find a broad coverage of diverse topics across the different semantic markup languages.

Chapter 7: Profiling of schema.org Microdata. This chapter focuses on the analyzes of the quality of semantic annotations in terms of schema compliance and duplicate detection. Insights about the quality are useful for any kind of subsequent usage of data. The chapter introduces a cleansing pipeline for semantic annotations using a set of heuristics to overcome the schema violations. The pipeline further removes duplicates within semantic annotations. The chapter shows that those two cleansing approaches dramatically influence the number of *uniquely* described entities. In some cases, the number is reduced by over 50%.

Chapter 8: Evolution of the Deployment of schema.org Microdata over Time. Based on the detection of the increasing spread of semantic annotations embedded byMicrodata with schema.org this chapter investigates the reasons and the mechanisms of this evolution in more detail. Such an analysis can help to estimate if valuable changes will be adopted in near future. Making use of a novel, data-driven approach, the chapter evaluates the influence of changes within the definition of the

vocabulary towards the the actual deployment and the other way around. Furthermore, studies are carried out to analyze the influence of incentives, provided by the main drivers of schema.org, on the deployment of promoted classes. The findings underline that the deployment can be encourage by incentives which also increase the consistency of semantic annotations in HTML pages over time.

PART III: Use Case-Specific Profiling.

The final part of the thesis moves the focus of the analysis towards an use case-specific profiling of semantic annotations. The main goal is to evaluate the utility of semantic annotations for specific use cases. In a first step, identifying the information need of three different use cases, the thesis shows the capability of semantic annotations to satisfy these needs. Making use of use case-specific methods a more fine-grained profile of product-related semantic annotations is generated, in a second step.

Chapter 9: Use case-specific Utility Analysis of schema.org Data. This chapter exemplary selects three use cases, namely a marketplace, a news aggregator, and a travel portal in order to analyze the utility of semantic annotations embedded by Microdata together with schema.org. Based on defined information requirements, e.g., certain properties which need to be provided by use case-related web sites, we show that the utility of the dataspace heavily depends on the use case. We find in particular that for the first two use cases a large fraction of the necessary properties can be gathered from various source. In the case of the more complex use case of the travel portal the related sources do not provide sufficient information to satisfy the defined data requirements.

Chapter 10: Product Data Categorization. In order to identify the topical distributions of the product-related semantic annotations, this chapter analyzes how web site-specific categorization information can be exploited to match the described products to a common product catalog. As so far, most related work makes us of a manually labeled dataset to train a classification model. Omitting this manual work increases the level of automatism, as well as reduce the need for resources to generate this labeled dataset. The results of the evaluation show that to a certain extend those information can be helpful but distant-supervised methods perform around 30% worse than supervised ones. This indicates, that at least for this specific task, manual work is necessary to integrate the product-related data as well as to generate a more fine-grained topical profile.

Chapter 11: Conclusion. The final chapter of the thesis summarizes the core contributions of the different chapters. The contributions are discussed with respect to the overall goal of the thesis. Furthermore, an overview of the research impact of the contributions is given. It is shown, that during the writing of the thesis the research community has started to recognize the potential of semantic annotations.

1.4 Published Work

Parts of the work presented in this thesis have been previously published in international journals and proceedings of international conferences:

International Journals:

- Robert Meusel, Dominique Ritze, Heiko Paulheim: *Towards More Accurate Statistical Profiling of Deployed schema.org Microdata*. Special Issue on *Web Data Quality* of the *ACM Journal on Data and Information Quality* (JDIQ), Vol 8, No 1, pages 3:1–3:31, 2016.
- Robert Meusel, Sebastiano Vigna, Oliver Lehmberg, Christian Bizer: *The Graph Structure in the Web – Analyzed on Different Aggregation Levels*. The Journal of Web Science, Vol 1, No 1, pages 33–47, 2015.

International Conferences:

- Robert Meusel, Anna Primpeli, Christian Meilicke, Heiko Paulheim, and Christian Bizer: *Exploiting Microdata Annotations to Consistently Categorize Product Offers at Web Scale*. Proceedings of the 16th International Conference on Electronic Commerce and Web Technologies (EC-Web 2015/T2), Valencia, Spain, September, 2015.
- Robert Meusel, Christian Bizer, Heiko Paulheim: *A Web-scale Study of the Adoption and Evolution of the schema.org Vocabulary over Time*. Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics (WIMS 2015), Limassol, Cyprus, July 2015.
- Robert Meusel and Heiko Paulheim: *Heuristics for Fixing Common Errors in Deployed schema.org Microdata*. Proceedings of the 12th Extended Semantic Web Conference (ESWC 2015), Portoroz, Slovenia, May 2015.
- Robert Meusel, Peter Mika, Roi Blanco: *Focused Crawling for Structured Data*. Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM 2014), Shanghai, China, November 2014
- Oliver Lehmberg, Robert Meusel, Christian Bizer: *Graph Structure in the Web – Aggregated by Pay-Level Domain*. Proceedings of the ACM Web Science 2014 Conference (WebSci 2014), Bloomington, USA, June 2014.
- Robert Meusel, Sebastiano Vigna, Oliver Lehmberg, Christian Bizer: *Graph Structure in the Web – Revisited*. Proceedings of the 23rd International World Wide Web Conference (WWW 2014), Web Science Track, Seoul, Korea, April 2014.
- Robert Meusel, Petar Petrovski, Christian Bizer: *The WebDataCommons Microdata, RDFa and Microformat Dataset Series*. Proceedings of the 13th International Semantic Web Conference (ISWC 2014), RBDS Track, Trentino, Italy, October 2014.

- Christian Bizer, Kai Eckert, Robert Meusel, Hannes Mühleisen, Michael Schuhmacher, Johanna Völker: *Deployment of RDFa, Microdata, and Microformats on the Web – A Quantitative Analysis*. Proceedings of the 12th International Semantic Web Conference (ISWC 2013), Sydney, Australia, October 2013.

International Workshops:

- Robert Meusel and Heiko Paulheim: *Creating Large-scale Training and Test Corpora for Extracting Structured Data from the Web*. Proceedings of the Third International Workshop on Linked Data for Information Extraction (LD4IE ISWC 2015), Bethlehem, Pennsylvania, USA, October, 2015.
- Robert Meusel, Blerina Spahiu, Christian Bizer, Heiko Paulheim: *Towards Automatic Topical Classification of LOD Datasets*. Proceedings of the WWW2015 Workshop on Linked Data on the Web (LDOW WWW 2015), Florence, Italy, May 2015.
- Robert Meusel and Heiko Paulheim: *Linked Data for Information Extraction Challenge 2014 - Tasks and Results*. Proceedings of the Second International Workshop on Linked Data for Information Extraction (LD4IE ISWC 2014), Trentino, Italy, October 2014.

Chapter 2

Preliminaries

This chapter first describes the realization of semantic annotations in HTML pages from a technical point of view. As briefly mentioned before, semantic annotations are embedded in the HTML code by extending the standard HTML markup with a set of additional attributes or class definitions. In the following those HTML extensions are referred to as semantic markup languages or markup languages for semantic annotations. Subsequent, this chapter introduces major vocabularies which are used together with the semantic markup languages in order to infuse a meaning to the encapsulated information. Subsequent we explain how semantic annotations are extracted from the HTML code and how they are stored for further processing. In the last section of this chapter, we discuss why we consider semantic annotations as own dataspace in the subsequent chapters.

2.1 Semantic Markup Languages

Currently, the three semantic markup languages Microformats, RDFa, and Microdata are most commonly used in the public Web. In order to describe the characteristics of those markup languages, we make use of the following HTML snippet as running example. The snippet, shown in Listing 2.1, contains information about a person.

Listing 2.1: Plain HTML Example Snippet.

```
1 <div>
2   <div>Max Mustermann</div>
3   <div>Google Inc.</div>
4   <div>01234/56789</div>
5   <a href="http://max.com/">Max Page</a>
6 </div>
```

From the given example snippet, a human may understand the content of the snippet, but a machine would need to guess (e.g., based on pre-trained model) what kind of information (e.g., a person) is described and which `<div>`-tag includes what kind of information.

2.1.1 Microformats

Microformats[9] (MF) are one of the oldest markup languages for semantic annotations in web pages. Technically, Microformats do not extend the standard set of HTML attributes but define a fixed set of values for existing HTML attributes. The most popular Microformats are listed in Table 2.1 together with their intended topical domains. Most of the listed Microformats, define CSS `class` attribute values in order to indicate in the HTML code the presence of the corresponding entity (e.g., an event). The root class names, indicating this presence, are listed in the most right column of the table. Only MF`XFN` does not define `class` attribute values but `rel` attribute values. This particular Microformat is used to define relationships between persons and therefore intended to be used with e.g., MF`hCard`. The list of all defined attributes values for the different Microformats can be found on the Microformats web site.

Table 2.1: List of different Microformats and their topical domain.

Microformats	Topical Domain	Root Class Name(s)
MF`geo`	locations	`geo`
MF`hCard`	people, contacts and organization	`vcard`
MF`hCalendar`	calendars and events	`vcalendar`, `vevent`
MF`hListing`	listings for products or services	`hlisting`
MF`hRecipe`	cooking and baking recipes	`hrecipe`
MF`hResume`	resumes and CVs	`hresume`
MF`hReview`	reviews and ratings	`hreview`
MF`species`	species	`species`
MF`XFN`	social relationships	defines only `rel` attribute values

Annotating the example HTML snippet from before using Microformats (in particular MF`hCard`) results in the code shown in Listing 2.2.

Listing 2.2: Microformats annotated HTML Example Snippet.

```
1 <div class="vcard">
2   <div class="fn">Max Mustermann</div>
3   <div class="org vcard">
4     <span class="fn">Google Inc.</span>
5   </div>
6   <div class="tel">01234/56789</div>
7   <a class="url" href="http://max.com/">Max Page</a>
8 </div>
```

The set of class attributes defined by Microformats allows a machine to understand that within the content of the first `<div>` a person is described (as indicated by the class attribute `vcard`). Further, the different values stated in the subsequent `<div>`s, name (`fn`), telephone number (`tel`) and the link to the homepage (`url`), can be programmatically interpreted. In order to annotate the organization (`org`), multiple values for the same class attribute are necessary (line 3) as the relation between the organization and the encapsulating entity needs to be defined. The

[9] `http://microformats.org/`

mechanism makes it somehow difficult to read the the annotation as Microformats do not differentiate between class and property values. In addition, due to the fact that Microformats define a set of values for HTML attributes, they cannot be combined with other vocabularies. Consequently, the topics which can be annotated are limited to the topical coverage of the values defined by the different Microformats attribute value sets.

2.1.2 RDFa

RDFa (Resource Description Framework in Attributes)[10] was first proposed in 2004 for the purpose of extending (X)HTML in order to support RDF. In 2008, RDFa became an official recommended standard by the *World Wide Web Consortium* (W3C) [Adida and Birbeck, 2008]. The set of attributes provided through RDFa focus on the (almost) direct inclusion of RDF into HTML code. As result, the defined attributes stick closely to the RDF syntax. In the RDFa subset *RDFa Lite* the attributes `vocab`, `prefix`, `resource`, `property`, and `typeof` are defined. Using those attributes almost all RDF expressions can be modeled in HTML. Furthermore, the full standard includes the additional attributes: `about`, `content`, `datatype`, `inlist`, `rel`, and `rev`. In order to include semantic annotations using RDFa, a (common) vocabulary is necessary which is further used to describe the meaning of the annotated information. Basically each existing vocabulary can be used.

The semantically annotated example HTML snippet using RDFa together with the vocabulary schema.org is shown in Listing 2.3.

Listing 2.3: RDFa annotated HTML Example Snippet.

```
1 <div vocab="http://schema.org/" typeof="Person">
2   <div property="name">Max Mustermann</div>
3   <div property="affiliation" typeof="Organization">
4     <span property="name">Google Inc.</span>
5   </div>
6   <div property="telephone">01234/56789</div>
7   <a property="url" href="http://max.com/">Max Page</a>
8 </div>
```

In contrast to the former example using Microformats, the code looks slightly more complex and might be harder to understand for humans, but allows a richer annotation, as it is independent of the pre-defined attributes values. Furthermore, the annotation of nested entities is more intuitive, as RDFa explicitly differentiates between classes and properties. In the example an `Organization` is defined, being the `affiliation` of the annotated `Person`. RDFa also allows the combination of different vocabularies whenever necessary as well as the definition of multiple types for the same entity.

[10] `http://www.w3.org/TR/xhtml-rdfa-primer/`

2.1.3 Microdata

Similar to RDFa, Microdata[11] [Hickson, 2011] also defines an additional set of attributes as an extension of the standard set of attributes defined within HTML. The first public draft for this extension was filed in 2008 with the purpose of allowing an easy embedding of machine-readable data in HTML documents.[12] The set of additional attributes can also be used together with a vocabulary to semantically annotate objects (in the context of Microdata called *items*) contained in code of a HTML page. The set of additional attributes consists of `itemscope`, `itemtype`, `itemprop`, `itemid`, and `itemref`.

Listing 2.4 shows the running example HTML snippet annotated with Microdata and the schema.org vocabulary.

Listing 2.4: Microdata annotated HTML Example Snippet.

```
<div itemscope itemtype="http://schema.org/Person">
  <div itemprop="name">Max Mustermann</div>
  <div itemprop="affiliation" itemscope
         itemtype="http://schema.org/Organization">
    <span itemprop="name">Google Inc.</span>
  </div>
  <div itemprop="telephone">01234/56789</div>
  <a itemprop="url" href="http://max.com/">Max Page</a>
</div>
```

In contrast to RDFa, Microdata defines less attributes, which makes it slightly easier to understand the code and also reduces the chance of errors during the annotations process, which follows the original intent of the semantic markup language. Furthermore, Microdata is designed to annotate each piece of information within a web page with a type of the same vocabulary, where the properties can originate from different vocabularies.[13] Similar to RDFa, Microdata also allows the annotation of nested information.

2.2 Common Vocabularies

As said before, RDFa and Microdata allow inclusions of semantic annotations in HTML pages solely in combination with a vocabulary. Therefore, we introduce the two most common used vocabularies within the context of the public Web the *Open Graph Protocol* as well as *schema.org*, and briefly mention other vocabularies, which are used in the Web.

As for now, a vocabulary is a collection of terms (classes and properties) which defines a meaning for one or more topics. For reasons of simplicity, we omit the

[11] `https://www.w3.org/TR/microdata/`

[12] `https://www.w3.org/standards/history/microdata`

[13] `https://www.w3.org/wiki/Mixing_HTML_Data_Formats#Mixing_Vocabularies_in_microdata`

distinction between vocabulary, ontology, and taxonomy and refer to them as vocabulary. Optionally, this vocabulary can define – in an ontology-like way – additional constraints such as domain and range definitions, and sub- and superclasses, which is referred to as the schema of the vocabulary.

2.2.1 Open Graph Protocol

The main idea behind the Open Graph Protocol (OGP) is to overcome the gap of other existing vocabularies in order to represent *rich objects within a social graph*. Therefore, the vocabulary focuses on simplicity for the end-user, in this case the data providers (web sites). The classes covered by OGP (referred to as *object types* in the context of OGP) are partly grouped in so-called *verticals*, containing classes of the topical domain *music*: `song`, `album`, `playlist`, and `radio_station`, as well as the topical domain *video*: `movie`, `episode`, `tv_show`, and `other`. In addition, the classes `article`, `book`, and `profile` are defined. Furthermore, the vocabulary defines four mandatory properties to describe an object from a web page within the open graph, which are: the `title` of the object, the `type` of the object, e.g. a video, a URL to an `image` representing the object in the graph, as well as a URL representing a permanent `link` to identify the object. Besides those mandatory properties, the vocabulary defines a larger set of optional, object type-specific properties such as a description or a link to an audio file.[14]

OGP Applications One of the main driver behind OGP is the social network *Facebook*. The network uses OGP as successor of their *Facebook Connect* to enable developers to access the *Facebook API*.[15] Using this service, developers can access objects within the Facebook ecosystem as well as publish objects, such as *articles*, *videos*, *music* and *movies* from outside of Facebook within the social network. A prominent use case is the enabling social plugins, e.g. the so called *Like Button*. Using this functionalities, Facebook participants can indicate that they *like* the objects (e.g, video or article) within Facebook. As with OGP also objects from outside of Facebook are integrated in the social network, those can also be liked and commented. A major motivation for data providers to connect their objects to the Facebook ecosystem is the increased visibility provided by the social network. Based on the 2015 report, Facebook has over 1.59 billion active users each month.[16]

[14] A full set of the available properties as well as defined types can be found at the Open Graph Protocol site: `http://ogp.me`. Note, that earlier the web sites `http://opengraphprotocol.org/schema/` was used to describe the vocabulary.

[15] General information about the API can be found on the Facebook developers page: `https://developers.facebook.com/`.

[16] Based on the 2015 full year report of Facebook `http://investor.fb.com/releasedetail.cfm?ReleaseID=952040`.

2.2.2 Schema.org

In June 2011, the schema.org initiative has started to create, maintain, and promote the schema.org vocabulary for semantic annotations within the Web. The initiative is sponsored by the search engine companies Bing, Google, Yahoo!, and Yandex, while the work is community driven. Everybody can initiate changes like additions, corrections or deletions to the current version through a public mailing list.[17]

The main goal of schema.org is to create a vocabulary, which, in comparison to existing once, provides "a single schema across a wide range of topics that included people, places, events, products, offers, and so on" [Guha et al., 2015].

As inspiration, `data-vocabulary`[18] was used and is therefore also known as the predecessor of schema.org. In the beginning, the vocabulary contained 297 different classes and 187 properties under the namespace `schema.org`. Till 2015, over 20 different releases have been published, which makes schema.org one of the most frequently updated vocabularies deployed in the Web.[19] Across those releases, the number of classes has doubled to 638 and the number of properties has increased by factor four to 965.

The starburst diagram shown in Figure 2.1 demonstrates the hierarchical structure of the schema.org classes, defined in version `2.2` which was released in November 2015. The inner gray circle represents the root class `Thing`. The subsequent circles represent the classes and their subclasses. From this figure, we can derive a maximum type hierarchy depth of six. Furthermore, marked by different colors, we can see that all classes are categorizes by one of the eight topical domains.

In contrast to OGP, the topical diversity of types within schema.org is broader and they are organized within a multilevel hierarchy. Both vocabularies make use of domain and range definitions for attributes, where the number of defined properties in schema.org is much larger.

Schema.org Applications The inclusion of semantic annotations in HTML pages using schema.org vocabulary allows for example search engine companies to directly understand the meaning of the contained information. For pages where such an embedding is missing, the meaning of the information has to be guessed (e.g., using classification models) or specific attributes (e.g., the price or the size of a product) could not be detected. But with the increasing number of web sites annotating (parts of) their data, the companies could directly make use of the information and create new services or improve existing services.

Based on [Guha et al., 2015], in 2011, as already mentioned before, *Google's Rich Snippets* where the first service making (partially) use of those information. In

[17] `https://github.com/schemaorg/schemaorg/issues`

[18] Unfortunately the web site of `data-vocabulary` of today directs the visitor to the web site of schema.org. An example definition of the type organization can be found on the archived page of `http://web.archive.org`: `http://web.archive.org/web/20100907193012/http://www.data-vocabulary.org/Organization/`.

[19] `http://schema.org/docs/releases.html`

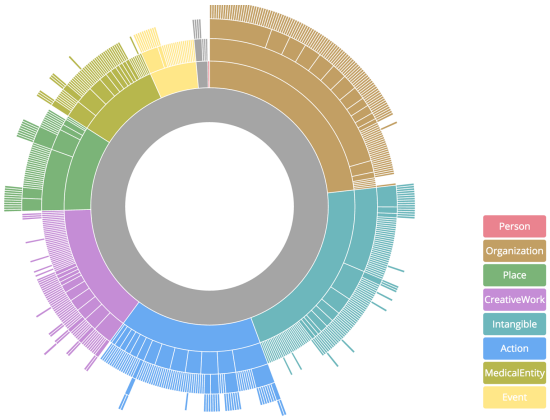


Figure 2.1: Starburst visualization of the schema.org's hierarchy (version `2.2`), adapted from [Brickley, 2015].

addition, *Google* makes use of schema.org markup with emails, especially within confirmations for reservations or bookings (hotels, restaurants, airlines). Here the email assistant application can understand the correlating appointments and use the information for notifications and reminders.[20] Besides, *Cortana* by *Microsoft* makes use of schema.org information in a similar way.[21] The operating system iOS 9 by *Apple Inc.* exploits semantic annotations to improve their search features [Guha et al., 2015]. Although the companies mention some of the applications using semantic annotations in their web master manuals, the list of application might be incomplete.[22] Furthermore, it is unknown how the data is preprocessed and whether the data is enriched with data from other sources like the *knowledge graph*.

2.2.3 Other Vocabularies

Besides the two mentioned vocabularies, which are recommended by major search engine companies and social network companies, also other vocabularies exist. In the following, we briefly discuss the ones which are also used commonly in the public Web. Most of those vocabularies were curated in order to allow the description of objects and processes within a more or less specific topical domain like libraries or relationships between individuals. Furthermore, they mostly existed already before the semantic annotations through RDFa and Microdata have been introduced and are partly not originally intended to be used within HTML pages.

[20] Further information on the integration of schema.org in gmail can be found on the web page: `https://developers.google.com/gmail/markup/`.

[21] `https://msdn.microsoft.com/en-us/library/dn632191.aspx`

[22] The web master manual of Google can be found on the following web page `https://developers.google.com/search/docs/guides/intro-structured-data`. The one by Bing is available on this web page `https://www.bing.com/webmaster/help/marking-up-your-site-with-structured-data-3a93e731`.

data-vocabulary[23] This vocabulary, also named the predecessor of schema.org, covers similar to schema.org a large set of topics of entities which can be found within the Web. Although the vocabulary is still used, the web site which used to specify the vocabulary references to the web site of schema.org.

Friend of a Friend (FoaF)[24] This vocabulary is designed in order to model the activities of people and the relations to other people [Graves et al., 2007]. Overall it includes all necessary terms in order to reproduce most interactions and relations going on in *social networks*.

Good Relations[25] This vocabulary was originally designed and introduced by Martin Hepp [Hepp, 2008]. The vocabulary focuses on the modeling and description of objects and activities related to the e-commerce domain. Good Relations was integrated into schema.org in 2012 (Release `0.99`) but is still further maintained independently.

Semantically-Interlinked Online Communities (SIOC)[26] This vocabulary covers classes and properties that are necessary to describe all kinds of online communication platforms like web blogs, message boards, wikipedias [Breslin et al., 2005]. In addition to the SIOC core vocabulary, the SIOC types module contains different possible sub-classes of the main web community concepts such as `BlogPost` and `Comments` for the core class `Post`.

Simple Knowledge Organization System (SKOS)[27] The classes and properties covered by this vocabulary focus on the representation and organization of knowledge organization systems (KOS) like topic maps, ontologies, thesauri and classification schema [Miles et al., 2005].

Dublin Core (DC) Terms & Elements[28] This vocabulary is a collection of standardized terms in order to describe documents within the Web [Weibel, 1997]. The vocabulary supports the annotation of meta information about those documents like the publisher, the year the document was created, and the authors, to name just a few. The initiative maintaining the vocabulary is mainly driven by libraries in order to create a generally valid standard to mark meta information.

[23] Till 2012, the specification of this vocabulary was available at the website `http://www.data-vocabulary.org/`, which now mainly references the website of its successor schema.org.

[24] The specification (as state of March 2016 Version 0.99) can be found on the website of xmlns: `http://xmlns.com/foaf/spec/`.

[25] `http://www.heppnetz.de/projects/goodrelations/`

[26] The specification for SIOC can be found at the web site of RDFS: `http://rdfs.org/sioc/spec/`.

[27] The SKOS specification as well as additional information can be found at the web site of W3C: `https://www.w3.org/2004/02/skos/`.

[28] `http://dublincore.org/`

Creative Commons[29] The vocabulary provides a set of copyright licenses to enable easy sharing and publishing of creative content within the Web. In order to annotate licenses within HTML, they provide a vocabulary defining the necessary information to do so.

2.2.4 Namespaces and Abbreviations

All of the before mentioned vocabularies make use of one or more namespaces when deploying terms of the vocabulary. Table 2.2 lists the most common namespaces for each vocabulary and states the abbreviation which we use in the following chapters to refer to the namespace or vocabulary.[30] Besides the already mentioned vocabularies, the table includes Facebook specific entries as the social network used to deploy various namespaces within their integration examples in the past.

Table 2.2: Overview of namespaces and abbreviations of selected vocabularies.

Vocabulary	Namespace	Abbreviation
OGP	http://ogp.me/ns#	ogm
	http://opengraphprotocol.org/schema/	ogo
Facebook OGP	http://www.facebook.com/2008/	fb2008
	http://ogp.me/ns/fb#	ogp/fb
schema.org	http://schema.org/	s
data-vocabulary	http://rdf.data-vocabulary.org/#	dv
Fried of a Friend	http://xmlns.com/foaf/0.1/	foaf
Good Relations	http://purl.org/goodrelations/v1#	gr
SIOC Core	http://rdfs.org/sioc/ns#	sioc
SIOC Types	http://rdfs.org/sioc/types#	sioctypes
SKOS	http://www.w3.org/2004/02/skos/core#	skos
Dublic Core Terms	http://purl.org/dc/terms/	dcterm
Dublic Core Elements	http://purl.org/dc/elements/1.1/	dc
Creative Commons	http://creativecommons.org/ns#	cc

2.3 Parsing Semantic Annotations to RDF

This section describes the data model (Resource Description Framework) which is used to store and further process semantic annotations. We also explain what technique (Apache Any23 Library) is used to detect and extract semantic annotations from the HTML code of web page within the context of this thesis.

Resource Description Framework[31] (RDF) is an universal mechanism to describe any kind of *resource* (collection of information about an object). It became a W3C standard in 2004 and was originally designed as a standard mechanism for meta data. Each elementary statement (one particular information about an object) is

[29] The interested reader can find more information on the creative commons web site about the different licenses as well as the provided vocabulary: http://creativecommons.org/.

[30] Most of those abbreviations are based on the suggestions by http://prefix.cc/.

[31] http://www.w3.org/RDF/

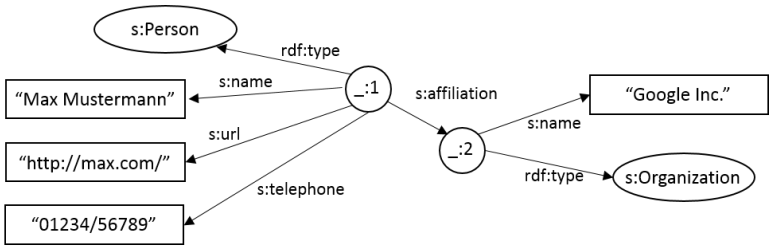


Figure 2.2: RDF Graph representation of semantic annotations in the example HTML snippet.

described through the combination of a *subject*, *predicate*, and *object* (referred to as *triple*) in RDF. Within the triple the resource (subject) is connected to another resource or simply a value (literal). The connection is further defined by the value of the predicate. The set of these triples which describe a resource can be represented as directed graph. In this graph, each arc and its associated nodes describe one triple (also called *statement*).

Figure 2.2 shows the graph representation of the information contained in our example HTML snippet annotated using Microdata (compare Listing 2.4). The circles represent resources, where the ovals represent the class or type of the resource. Rectangles are used to represent literal values.

In our example, the nodes of the resources (`_:1` and `_:2`) are so called *blank nodes* as no URI or literal is given for the resources, which is mostly common in semantic annotations. The type/class of the resources is described by the property `rdf:type`. Besides the graph representation, the information can also be represented as a set of triples:

```
_:1 rdf:type s:Person .
_:1 s:name "Max Mustermann" .
_:1 s:telephone  "01234/56789" .
_:1 s:url "http://max.com/" .
_:1 s:affiliation _:2 .
_:2 rdf:type s:Organization .
_:2 s:name "Google Inc." .
```

Within this thesis, most of the introduced algorithms and methods make use of the such a line-based input format rather than the graph representation.

Apache Any23 Library[32] (Any23) is a Java library provided by the *Apache Any23 Project*. Any23 includes methods which can locate and extract embedded semantic annotations by the three semantic markup languages, as long as the markup language-specific syntax is correctly included in the HTML page. Hence, the

[32] `https://any23.apache.org/`

surrounding HTML markup does not need to be valid in order to extract the information. Any23 does not do any cross-checks between the vocabulary definition (e.g., domain and range definitions) and the annotated information. Meaning, that as long as the markup language is correctly used all semantic annotations are extracted, independed of the used vocabulary, and the correct or incorrect spelling of classes and properties. The library reconstructs the data as RDF, where we extend the triple-based representation by the URL of the web page the triple was extracted from and store the resulting quads as *n-quads*[33].

2.4 Dataspaces

The term *dataspace* was introduced by [Franklin et al., 2005] in 2005, as an attempt to describe the collection of "a large number of diverse, interrelated data sources" with respect to the purpose of data integration and management. As logical components of a dataspace the authors identify *participants* and *relationships*. As participants, they define any kind of individual data source such as databases or XML repositories. Those participants provide structured, semi-structured or unstructured data through a more or less expressive interface (such as web services or structured files). Furthermore, relationships are defined as any relation between multiple participants (P_1, P_2 .. P_n), such as P_1 being a subset of P_2, or P_1 and P_2 are maintained by the same organization.

Based on this definition, one can argue that the Web itself is a dataspace, where the data providers (the administrative authority of the web pages belonging to one web site) can be seen as the *participants*, providing data e.g., through HTML pages, web application programming interfaces or File Transfer Protocol (FTP) servers. Especially in the light of this thesis, data provided through HTML pages annotated with the techniques described before can be seen as subset of this *web dataspace*. Hence, its also an own dataspace. Within this particular dataspace, we can establish topical relations whenever two or more participants provide data about a similar topic. In addition, data provided by a participant can (partly) be a copy of data provided by other participants. Therefore, we will make use of the term dataspace in the context of this thesis whenever we talk about the collection of data provided through HTML pages using semantic markup languages (together with a vocabulary). Whenever we reduce the object of study to one specific markup language, such as Microdata together with a specific vocabulary, such as schema.org, we more specifically refer to this as the *schema.org Microdata dataspace*.

Besides the term dataspace, also the term *data lake* has become more and more popular in the area of data integration and data management. The term was first mentioned by [Dixon, 2010] in 2010 and is referred to as a repository to store data in its raw format and with its original data schema ignoring, in a first step, how and for what purpose the data is later used. More recent industry-related definitions also talk about data lakes in combination with the ability to

[33] `https://www.w3.org/TR/n-quads/#n-quads-language`

perform analyzes of the contained data like the *Microsoft Azure Data Lake*[34], which internally builds up on the *Apache Hadoop* stack. This definition shifts data lakes towards the direction of data warehousing applications which are capable of transforming, integrating and preparing the data for further analysis or data mining tasks [Vaisman and Zimnyi, 2014].

Nevertheless, although no official definition of data lakes exists, it is more commonly used in order to refer to the technical enabling of various applications to work on data from diverse sources and in different formats. As the scope of this thesis does not lie on the technical realization of a data integration platform, we do not consider the term data lake to describe the collection of semantic annotations collected from various web sites.

[34] https://azure.microsoft.com/en-us/solutions/data-lake/

Part I

Extraction of Semantic Annotations

Chapter 3

Data Extraction from the Web using Focused Crawling

In order to perform empirical studies on the deployment of semantic annotations in the Web, we first need to collect a sufficient large amount of such annotations from web pages. semantic annotations are not provided via a separated interface like a web service or an application programming interface by the web sites. Consequently, it is necessary to search for them in the whole Web by inspecting the content of the each web pages individually.

Classically *crawlers* (also called *spiders*) are used to collect any kind of data from the Web. In theory, such applications can navigate their way through the whole Web and harvest the content of visited web pages. More precisely, crawlers can only reach web pages which are connected to other web pages by hyperlinks. General crawlers select the web page which should be visited next based on the order the web pages were discovered or based on the popularity of the web pages.

In contrast, *focused crawlers* (also called *directed crawlers*) are designed to collect the content of web pages based on a predefined objective function. In the past, these kind of crawlers have been used to discover web pages containing information about a specific topic (e.g., medical information). State-of-the-art focused crawling approaches employ classification models, using various sets of features, to decide whether a newly discovered web page should be visited or not. In particular, the use of so called online classification approaches, which continuously try to improve the trained classification model has shown promising results.

In the following, the thesis therefore evaluates to which extend this strategy can be adopted to collect semantic annotations from the Web. The state-of-the-art strategies are extend by a *bandit-based selection* in order include a higher flexibility between the exploration of new web pages and the exploitation of familiar web pages. The use of focused crawling for the collection of semantic annotations is somehow different, as not the described information on the web page is objective of the crawling but the deployed semantic markup languages.

In order to provide an overview of crawling and in particular focused crawling the next section summarizes the most relevant work in this area. Before actually introducing specific crawling techniques, the section summarizes research trying to describe the overall structure of the Web. This knowledge is essential in order to understand the different challenges of crawling. A detailed description of the strategy of focused crawling and the extensions which are implemented in comparison to state-of-the-art approaches are described in Section 3.2. The experiments and their results, which we performed in order to evaluate the approach for the use case of crawling semantic annotations are presented in Section 3.3 and Section 3.4. The chapter summarizes and discusses the outcome in the final section.

The methodology as well as the evaluation of the results presented in this chapter have already been published in [Meusel et al., 2014a].

3.1 Related Work

As already mentioned before, (focused) crawlers are applications which are capable to navigate the Web. They follow hyperlinks, directing them to new web pages, which they discover within the HTML code of already visited web pages.

In order to deeply understand the challenges of crawling and its limitations this section first introduces related work describing the structure of the Web. Afterward, related work from the area of crawling and focused crawling is presented. For the task of focused crawling the section also discusses the techniques of online learning and bandit-based selection.

3.1.1 Structure of the Web

In general, the Web consists of different kinds of documents, such as HTML pages (identified by an uniform resource locator (URL)) and connections, such as hyperlinks (pointing to a specific URL) leading from one document to another. Consequently, the Web can be represented by a huge directed graph (web graph), where each node corresponds to a document within the Web and each arc (a directed edge) represents a hyperlink from one document to another.

The scale of the web graph; the number of nodes (e.g., web pages) and the number of arcs (e.g., hyperlinks) is rapidly changeable. One reason for this changeability is the way documents are created within the Web. Most web pages like those of huge shopping web sites such as *Amazon* are dynamically generated based on data of the underlying database. Whenever an entry in the database is added or removed a new web page is created or an existing web page is removed and hence a node (dis-)appears in the web graph. Consequently the related arcs of the node are added or removed.

Another reason for the changeability of the web graph arises from dynamic content as part of web pages. An example are advertisements contained in the

page which can differ between two visits of the same web page. Another example are personalized web pages which adapt the displayed content to the user-specific preferences like the *landing page* of `amazon.com`. Although it is difficult to determine the number of web pages in the Web, the number of web sites (also referred to as domains) in the Web can be estimated as all domains need to be registered.

As mentioned before, crawlers can only follow arcs, meaning they cannot follow them in the opposite direction. This is reasonable, as the crawler only knows the hyperlinks from the web page it just visited and cannot know all hyperlinks leading to the particular web page. Therefore, it is essential to carefully select the web pages where the crawler starts its exploration. This selection can already reduce the number of possible web pages which can be reached at maximum.

In order to select good starting points, the structure of the web graph needs to be considered. One of the first works, trying to get an imagination about the structure of the Web, was performed by [Broder et al., 2000]. Their contribution was twofold. First, as one of the first, they calculated for various distributions like the in- and out-degree of documents in the Web the most likely distribution. They found that almost everything follows a power-law. Their second contribution was to describe the overall structure of the Web as a *bow-tie*, claiming that over half of the Web is strongly connected [Dorogovtsev et al., 2001]. Nodes contained in such a strongly connected component are reachable from any other node in this component. Whereas weakly connected solely means that each node in the component is connected to at least one other node in the component. Hence, within a weakly component of a directed graph it is not possible to reach from each node any other node (due to the direction of the edges).

Figure 3.1 depicts this *bow-tie* structure, where the authors defined five different components to describe the overall picture:

LSCC All nodes within this main component of the Web form a large *strongly connected component* (SCC). Within this component, by definition, each node can reach by following the arcs each other node.

IN From nodes within this component, it is possible to directly or indirectly (by passing other nodes) reach nodes of the LSCC.

OUT Nodes within this component can be reached from nodes in the LSCC. But from this nodes it is not possible to reach any other component.

Tendrils & Tubes The first component includes nodes which can be reached from the IN component or which lead to the OUT component, without being part of the LSCC. Whenever a tendril starting from the IN component is connected to a tendril leading to the OUT component this combination is called a tube.

Disconnected This component includes nodes, which might be connected among themselves, but do not have any connection to the other components.

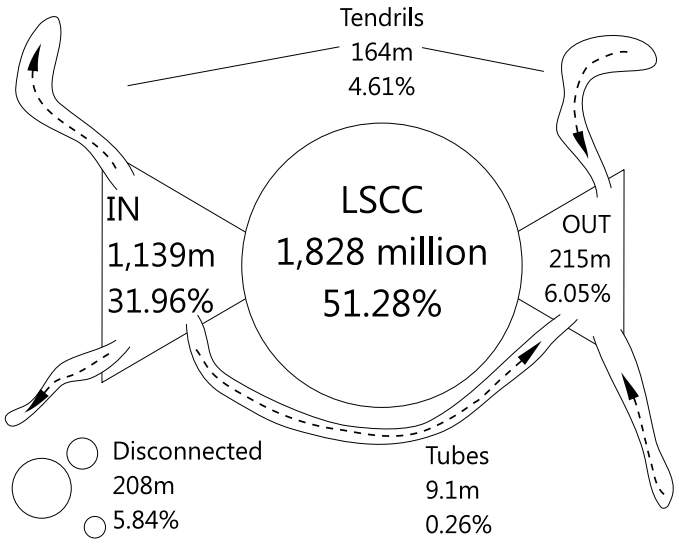


Figure 3.1: Bow-tie structure of the Web in 2012, adapted from [Broder et al., 2000] and [Meusel et al., 2014c].

Most of the findings have been proven to be (partly) incorrect in the following years. [Broder et al., 2000] solely used visual approaches to fit the distribution, where [Clauset et al., 2009] made use of more sophisticated statistical methods in order to calculate the likeliness of the in- and out-degree distribution to be a power-law and proofed that they do not fit. In one of our previous work, analyzing the so far largest public available web graph containing over 3.4 billion nodes, we have confirmed this finding [Meusel et al., 2015c, Meusel et al., 2014c, Lehmberg et al., 2014a]. Within the same works we also have relativized the findings about the structure of the Web. We confirmed that there is one giant strongly connected component, where each document can reach each other by following the hyperlinks. In addition to the SCC there are pages which lead (directly or indirectly) to the SCC and pages which are reachable from the SCC (referred to as IN and OUT respectively by [Broder et al., 2000]). But the sizes of those other component heavily depend on the used crawling strategy.

Besides our work and the work of Broder, other scientific studies have been published which mostly focus only on a part of the Web and/or making use of much smaller crawls. The work described in [Donato et al., 2005] and later in [Serrano et al., 2007], using multiple, general, but smaller web crawls, concluded also that the view on the structure of the Web is biased by the chosen crawling strategy. Other work, like [Boldi et al., 2002], [Baeza-Yates and Poblete, 2003], and [Zhu et al., 2008] focused on the Web of a particular geographic region (e.g., China or Africa). They found that the underlying graphs of the regional-restricted crawls show strong differences in comparison to the overall structure of the Web reported in [Broder et al., 2000].

Implications for Crawling

In the case of collecting (representative) snapshots of the Web knowledge about the general structure of the Web is an important source of information. First, the reachability of nodes from different components (IN, OUT, LSCC) suggests the use of web pages from the IN component to start the crawling (seeds), as otherwise some web pages cannot be reachable at all. Second, the number of newly discovered hyperlinks will increase strongly with each collected web page. As most crawlers are limited in time and resources and hence cannot follow all discovered hyperlinks a selection strategy is necessary to chose the most suitable page, which should be crawled next. Furthermore, during the process of crawling the same hyperlink will be discovered multiple times. Therefore, crawlers need to maintain a list of all crawled pages to omit the collection of the same page over and over.

3.1.2 Crawling

As we have discussed the implications of the structure of the Web for crawling, we move our focus towards the mechanisms of exploring and visiting documents (nodes) within the graph to collect data from their content. An overview of crawling in general is given in [Olston and Najork, 2010]. The major challenges in the area of crawling are discussed in [Cambazoglu and Baeza-Yates, 2011] and [Cambazoglu and Baeza-Yates, 2015].

As summarized in [Dhenakaran and Sambanthan, 2011], the overall behavior of crawlers is mainly influenced by four different policies: *selection*, *politeness*, *re-visiting* and *parallelization*. The *selection policy* (also called *selection strategy*) defines which of the newly discovered hyperlinks the crawler follows next. The most common selection strategies are breadth-first (search) (BF(S)) and PageRank [Page et al., 1999]. BF(S) is based on the order in which new arc to documents are discovered. PageRank, which is partially also used by the big search engine companies to steer their crawlers, measures the *importance* or *popularity* of a new arc to a document by checking how many arcs already have been found pointing to this document[35]. Especially the selection policy is adapted for the purpose of focused crawling. The *politeness policy* defines the frequency of request for web pages belonging to the same web site. This policy is essential in order to avoid been excluded (blocked) by a web sites.[36] The *re-visiting* is important whenever maintaining an up-to-date web corpora, as it defines the frequency of crawling the content of the same web page again. The last policy describes the capability of a crawler to distribute the collection of web pages among different threads (on the same or different machines). The capability to crawl multiple web pages at the same time directly positively effects the page collection rate.

[35] This explanation is rather trivial and should only illustrate the difference between the two selection strategies. For a deeper and more sophisticated explanation of PageRank please see [Page et al., 1999].

[36] This mechanism is mainly installed to prevent services from being shut down by DDOS attacks [Mirkovic and Reiher, 2004].

3.1.3 Focused Crawling

The main difference between crawlers in general and focused crawlers is the kind of *selection policy* which is employed. As mentioned before, this policy defines which newly discovered web page will be visited next by the crawler. An optimal selection strategy for a specific objective avoids the downloading of unnecessary web page in an ideal setup. For the case of collecting semantic annotations, the selection strategy only selects web pages which either contain semantic annotations or contain links to such web pages.

Focused crawling was first mentioned in [Menczer, 1997] who modeled the problem inspired by work on agents adapting to different environments. Two years later, [Chakrabarti et al., 1999] coined the term focused crawler and introduced an approach using a pre-trained classifier to assign topic-labels to new URLs based on features which could be extracted from the URL itself. Other classification features have been obtained using different natural language processing (NLP) techniques [Kan, 2004, Kan and Thi, 2005, Puurula, 2012]. Furthermore, [Diligenti et al., 2000] used information collected by web search engines in order to gather additional features for classification. [Aggarwal et al., 2001] incorporated information gathered during crawling to steer the direction of the crawler and maximize the number of retrieved relevant pages. They use features extracted from the content of the father of the page (i.e., the page where we found the link), retrieving tokens from unseen URL strings and features collected from sibling pages (i.e., whose URLs were discovered on the same page as the one to be crawled). After crawling a page, the probability of the different feature groups for a given *topic* is evaluated and the combined probability is used to update the priorities of unseen pages. Although this model makes use of features gathered during the crawling process, the probabilistic model needs to be manually adjusted beforehand, which [Chakrabarti et al., 2002b] try to overcome when first introducing an online classification approach for focused crawling. The authors crafted two classifiers, one static, pre-trained from an upfront collected and tagged corpus, and one online, which was used to improve former decisions based on features extracted from the document object model, e.g., the anchor text in links of crawled pages. Four years later, [Barbosa and Freire, 2007] took on the main idea of incorporating information gathered during crawling to steer the crawler with an extended feature set. The methodology is also referred to as *reinforcement learning* in literature. Besides the context of the page where a URL was found, they made use of the graph-structure of web pages, for example by distinguishing between direct features retrieved from the father and the siblings of the page, which was later also used in [Zheng et al., 2009]. Although they incorporate information gathered during crawling, they only replace their classifier with an updated version in batches, solely employing newly gathered information and discarding formerly extracted information. Their results indicate that sequentially updated classifiers lead to higher rates of gathering web forms for certain topical domains. [Umbrich et al., 2009] proposed a pattern-based approach to classify pages, in order to find specific media

types in the Web. [Jiang et al., 2012] used a similar method to learn URL patterns that lead to relevant pages in web forums.

The main difference from the approach we used with respect to mainstream focused crawling is that we are not aiming to perform topic-based classification, but rather looking at the value of web pages from the perspective of the data they contain. Web pages deploying semantic annotations have unique characteristics; semantic annotations are more common to particular types of pages, e.g., item detail pages, and favored by particular web sites, typically large dynamically generated sites serving certain types of content.

Our target is also distinct from that of native *semantic web* crawlers that collect documents in RDF document formats, which follow `seeAlso` and `sameAs` references to related data items in order to discover new linked data sources and information. Two examples are *Slug* [Dodds, 2006] and *LDSpider* [Isele et al., 2010]. A focused crawler for linked data was presented in [Yu et al., 2015] using relevance feedback based on a given set of seed entities in order to harvest the most relevant entities from the LOD Cloud. These crawlers deal with the specific issues related to RDF data on the Web such as support for various native RDF formats, supporting various communication protocols. In contrast, our work focuses on the harvesting of semantic annotations embedded inside HTML pages.

In the following, we explain in more detail state-of-the art online learning approaches as well as dealing with the explore/exploit problem in the light of focused crawling.

Online Learning for Focused Crawling

State-of-the-art focused crawlers partially make use of information gathered during crawling which is incorporated into the classification process in order to improve the accuracy of the prediction for unseen web pages. In contrast to the aforementioned works, we propose an online learning method that continuously obtains feedback during crawling and incorporates it directly in an online classifier, rather than replacing the classifier from time to time. Such methods have been used before whenever data is available as stream [Zliobaite et al., 2011] and the distribution of features within the data changes over time [Moreno-Torres et al., 2012]. Our underlying classification model makes use of all available feedback which can be successfully exploited from crawling for semantic annotations, independently from its topic, and gathers the largest number of relevant pages constrained to a given fetch budget.

Explore/Exploit for Focused Crawling

Existing crawling policies implemented in the systems above focus largely on maximizing the immediate reward available to the crawler and lack in the discovery of new pages which potentially lead to more other relevant pages, but do not contain relevant information directly [Dasgupta et al., 2007]. This problem can be described

as the trade-off between *exploitation*, the crawling of pages where the expected value can be predicted with a high confidence and *exploration*, the search for new sources of relevant pages [March, 1991, Kane and Alavi, 2007, Pant et al., 2002]. We address the issue of trading-off *exploitation* versus *exploration* by translating the problem of crawling into a *bandit* problem. We group newly discovered, not yet crawled web pages by their corresponding host, each representing one bandit. During each iteration, where we want to select a new page to be crawled, we either select a page from a bandit, whose expected gain for a given objective function is maximal (exploitation) or select a page from a randomly chosen bandit (exploration). This approach was analyzed using synthetic data by [Pandey et al., 2007] and successfully applied by [Li et al., 2010] in the context of news article recommendation. Its value for focused crawling has not been established before.

3.2 Focused Crawling Methodology

In the following, we present two commonly used approaches to machine learning (online classification and bandit-based selection) that we adapt to the domain of focused crawling, and in particular to the task of collecting semantic annotations from web pages.

3.2.1 Online Classification

Crawling pages that deploy semantic annotations can be cast as a focused crawling task, as their general aim is to devise an algorithm to gather as quickly as possible web pages relevant for a given objective function. Standard focused crawling approaches target pages that include information about a given topic, like *sports*, *politics*, *events* and so on. In our case, our primary objective function are web pages which make use of one of the three semantic markup languages Microformats, RDFa, and Microdata (compare Chapter 2), although there could be variants that narrow down this subset (compare Section 3.4.4).

Focused crawlers make use of topic-specific seeds and operate by training a classifier that is able to predict whether a newly discovered web page (before downloading and parsing its content) is relevant for the given target or not. Thus, it is mandatory to assemble a training set, find suitable topic seeds and learn a classifier *before* the crawling commences.

Online learning approaches adapt the underlying model used for classification on the fly with new labeled examples. In the case of a crawler, this would be suitable under the condition that it is possible to automatically acquire a label for a web page as soon as the content of the crawled page has been parsed. This approach is appealing because not only it is not necessary to create a training set in advanced but also the classifier adapts itself over time. In the case of the Web, where the distribution of single features is hard to predict it might happen that, while discovering larger amounts of pages the actual distribution differs strongly from the

one of the training set. This adaptability is useful to ensure suitable classification results [Moreno-Torres et al., 2012].

Feature Generation

In order to predict the relevance of an unseen newly discovered page it is necessary to extract features for each web page which are used by the classifier to make its prediction.

We considered three major sources of features which are (partly) available for a web page *before* downloading and parsing it:

1. *the URL*, which can be handled using NLP techniques to transform them into a feature vector.
2. *information coming from the parents of a page*, whose content has been already downloaded and the relevance for a given objective function is known.
3. *information coming from the siblings of a page*, meaning other pages which were found on the parent page, and whose relevance for a given target might already been known.

We note that these sources of features may become gradually available during the crawling process. We will always know the URL of candidate pages, but we might not have discovered *every* parent of a page; furthermore, information about siblings could not be available at all.

There are several possibilities to extract features from the URL of a page. In general, the URL is split into tokens whenever there is a non alphanumeric character (punctuation, slashes and so on) and these tokens can be directly used as features of the page. In order to reduce the sparseness of the generated feature vectors, and potentially improve the accuracy of the classifier, it is possible to apply several pre-processing steps for the extracted tokens before finally transforming them into features. Among common standard transformations [Baeza-Yates et al., 1999], such as lower casing and filtering of stop words, for example, we include removing tokens consisting of too few or too many characters. Another transformation maps different spellings of a given token into its normalized version, or replaces tokens only made up by numbers with one static token representing *any* number.

Dealing with Unknown Number of Features

Importantly, crawlers may not be aware of the range of different tokens (i.e., the dictionary) that can be extracted from the URL of newly discovered pages, which makes it difficult to use a pre-defined feature space for online learning. We overcome this problem by relying on the so-called *Hash-Trick* [Shi et al., 2009] and map all tokens into a fixed feature space.

This approach receives a list of pre-processed URL tokens previously split, $\{t\}$ and it creates a feature vector V with length k for the new page. First, it initializes every component with 0 values. Then, it maps each token t within the

list to $x_t \in [0..(k-1)]$ using the hash-function described in Equation 3.1, where n is the number of characters of t, k is the number of selected hashes and $t[i]$ is the numeric value of the character at position i.[37] The corresponding position within the feature vector will then be updated $V(x_t) \leftarrow 1$. This way, we can ensure that the number of features remains the constant during the whole crawling process. Although the known drawback of hash-functions is the potential information loss whenever a collision happens, the described approach achieved good results in our case, when hashing tokens from URLs.

$$x_t = \left|\left| \frac{\sum_{i=0}^{i<n} t[i] \cdot 31^{n-i+1}}{k} \right|\right| \tag{3.1}$$

We also extract features from the parents and siblings of a page. These features are based on labels assigned to parent/sibling pages previously. For example, we introduce as a feature the number of parents/siblings labeled with the target class, and a binary feature representing the existence of at least one parent or sibling labeled with the target class. Again it is important to state, that those information might not be available for all discovered pages. In case of multiple parents, not all need to be discovered and crawled already. The same applies for the sibling pages.

In the following section, we introduce the notion of bandit-based selection and explain how we combine online classification with this kind of selection strategy.

3.2.2 Bandit-Based Selection

A bandit-based selection approach estimates the relevance of a group of items for a given target, and performs this selection based on the expected gain (or relevance) of the groups. The bandit operates as follows: at each round t we have a set of actions $\mathcal{A}$ (or *arms*[38]), and we choose one of them $a_t \in \mathcal{A}$. Then, we observe a reward $r_{a,t}$, and the goal is to find a policy for selecting actions such as the cumulative reward over time is maximized. The main idea is that the algorithm can improve its arm-selection strategy over time with every new observation. It is important to note that the algorithm receives no feedback for unchosen arms $a \neq a_t$.

An ideal bandit would like to maximize the expected reward $max_a\, E(r|a, \theta^*)$, where θ^* is the true (unknown) parameter. If we just want to maximize the immediate reward (exploitation) we need to choose an action to maximize $E(r|a) = \int E(r|a, \theta)p(\theta|D)d\theta$, where D is the past set of observations (a, r_a). However, in an exploration/exploitation setting we want to randomly select an action a according to its probability of being Bayes-optimal

$$\int \mathcal{I}\left[E(r|a,\theta) = \max_{a'} E(r|a',\theta)\right] p(\theta|D)d\theta\ , \tag{3.2}$$

[37] The numerator is equal to the *hashCode*-function implemented for string objects in Java.

[38] Some bandits use *contextual* information as well [Li et al., 2010].

where $\mathcal{I}$ is the indicator function. In order to avoid computing this integral it suffices to draw a random parameter θ at each round t. One of the simplest and most straightforward algorithms is λ-greedy, where in each trial we first estimate the average payoff of each arm a. Then, with probability $1 - \lambda$, we choose the one with the highest payoff estimate $\hat{\theta}_{t,a}$ and with probability λ, we choose a random arm. In the limit, each arm will be tried infinitely often and the estimate $\hat{\theta}_{t,a}$ will converge to the true value θ_a. An adaptation of this straightforward algorithm is the use of a decaying λ_t. This adaptation faces the problem of coming up with a large number of random selection when the estimated $\hat{\theta}_{t,a}$ is close to the true value θ_a. A decaying λ_t approaches 0 faster with each iteration. We will later employ a linear decaying factor, $\lambda_t = \lambda \cdot \frac{m}{t+m}$, where m is a constant.

Adaptations for Crawling

In the case of our crawler, we use our bandit-based approach to make a first selection of the host to be crawled. This is motivated by the observation that the decision to use semantic annotations is performed at a host-level in most cases. Informally, we represent each host with a bandit that represents the value of all discovered pages belonging to this host. The available functions to calculate the score for a host and by this the estimated relevance for a target are diverse and described next. It is important to remark that selecting an arm (action) in this context would mean to select the host, which at a given point in time t has the highest expected value to include pages which are relevant for our target. Once we have selected the host, we follow by selecting a page from that host using the online classifier.

Formally speaking, each host $h \in H^t$ represents one possible arm, which can potentially be selected by the bandit at an iteration t. Each h includes a list of all pages p belonging to this host. An action $a_t \in \mathcal{A}$ within our approach is then defined as the selection a host $h \in H^t$ based the estimated parameter θ_h at a given t and λ. In order to estimate θ_h for an arm, we can think about various different combination of available features. Next we introduce the general approach and different functions to compute the score $s(h)$ using the following notation:

- $s(h)$ is defined as the score of the host h (or, in bandit notation, the expected reward $E(r|a, \theta^*)$).
- $C_{all,h}$ is the set of pages of h, which have already been crawled.
- $C_{good,h}$ (respectively $C_{bad,h}$) is the set of pages of h (not) belonging to the target class, which have already been crawled.
- R_h^t is the set of pages of h, which was already discovered but not yet crawled at iteration t. It is part of the set of pages in the bandit representing h.
- $pred(p)$ is defined as the confidence value of p to belong to the target class, based on the used classification approach.

Our general approach is to group all newly discovered pages into the corresponding host. To select a new page, we first use the bandit algorithm to identify the host of the page selecting the one with the current highest score or one random

(depending on the value of λ_t). From this selected host, we take the page with the highest confidence for the target class. This process is depicted in Algorithm 1.

Data: Initial back-off probability λ, initial seed set R_h, decaying factor m
$\lambda_t \leftarrow \lambda, C_{bad,h} \leftarrow \emptyset, C_{good,h} \leftarrow \emptyset \; \forall h \in R_h$
for $t \leftarrow 1$ **to** T **do**
  Draw uniformly a random number $n \in [0..1]$
  if $n > \lambda_t$ **then**
    for $h \in H^t$ **do**
      if $|R_h^t| > 0$ **then**
        Compute the score $s(h)$
      end
    end
    Select host $h = \text{argmax}_{h \in H^t} \, s(h)$
  else
    Select a random host h *where* $|R_h^t| > 0$
  end
  $p \leftarrow h = \text{argmax}_{p' \in R_h} \, pred(p')$
  crawl p *and observe reward* $r_{h,t}$
  if $r_h = 1$ **then**
    add p to $C_{good,h}$
  else
    add p to $C_{bad,h}$
  end
  update H *and* R_h *with new* p^*, h *retrieved from* p
  for $\forall$ *new* h **do**
    $C_{bad,h} \leftarrow \emptyset, C_{good,h} \leftarrow \emptyset$
  end
  $\lambda_t \leftarrow \lambda \cdot \frac{m}{t+m}$
end

Algorithm 1: Adapted general k-armed Bernoulli λ-greedy bandit for focused crawling, with a linear decaying factor.

Note that the bandit is unable to use single pages as *arms*, given that we only need to retrieve them once and the feedback loop would be rendered useless. We also classify per-host web page to prioritize them after a host is selected for crawling. A pure bandit-based approach would select a random page from within the host.

Scoring functions

As shown before, the selection of the next bandit is based on the score. In the case of our crawler, the score $s(h)$ can be calculated using different possible functions, which we define in the following:

Negative Absolute Bad function, where the score of a host is the negative number of already crawled pages not belonging to the target class of this host:
$s(h) = -|C_{bad,h}|$.

Best Score function, where the score of a host is defined by the maximal confidence for the target class of one of its containing pages:
$s(h) = \max_{p \in h} \text{pred}(p) \; \forall p \in R_h$.

Success Rate function, where the score of a host is defined by the ratio between the number of pages crawled, belonging to the target class and those not belonging to this class. The ratio is initialized with prior parameters α and β which we set both to 1:
$s(h) = \frac{C_{good,h}+\alpha}{C_{bad,h}+\beta}$.

Thompson Sampling function, where the score of a host is defined as a random number, drawn from a beta-distribution with prior parameters α and β. This function is based on the *k-armed Bernoulli bandit* approach introduced by [Chapelle and Li, 2011] and described in algorithm 1. In this case we take as the score at iteration t the random draw:
$s(h) = Beta(C_{good,h} + \alpha, C_{bad,h} + \beta)$.

Absolute Good · Best Score function, where the score is the product of the absolute number of already crawled relevant pages $|C_{good,h}|$ and the *best score function*:
$s(h) = |C_{good,h}| \cdot \max_{p \in h} \text{pred}(p) \; \forall p \in R_h$.

Thompson Sampling · Best Score function, where the score is the product of the *thompson sampling function* and the *best score function*:
$s(h) = Beta(C_{good,h} + \alpha, C_{bad,h} + \beta) \cdot \max_{p \in h} \text{pred}(p) \; \forall p \in R_h$.

Success Rate · Best Score function, where the score is the product of the *success rate function* and the *best score function*:
$s(h) = \frac{C_{good,h}+\alpha}{C_{bad,h}+\beta} \cdot \max_{p \in h} \text{pred}(p) \; \forall p \in R_h$.

Note that the reward depends on the target function of the bandit; in general we assign a positive reward only if the page crawled contains some form of markup data, but the process works similarly for other different objective functions (compare Section 3.4.4).

3.3 Experimental Setup

In this section we describe the architecture and the process flow implementing the methodology discussed in Section 3.2. The whole implementation as well as an integration into the Apache Nutch crawler framework[39] presented by [Khare et al., 2004] was publicly released [Ristoski et al., 2015] and is available within the Yahoo GitHub repository[40]. Furthermore, we introduce the dataset we used for the evaluation. We describe the different experiments we perform in order to measure the performance of our focused crawling approach for semantic annotations and compare it towards state-of-the art crawling approaches.

[39] `http://nutch.apache.org/`
[40] `https://github.com/yahoo/anthelion`

3.3.1 System Architecture and Process Flow

As input the application takes a queue of newly discovered, already filtered URLs[41] – named *input queue* Q_I. The output of the application is another queue where the URLs are ordered by the expected relevance for a given target – called *ready queue* Q_R.

URLs coming from Q_I are internally grouped by their host $h \in H$. Whenever a host h is selected, it is enqueued into the ready host queue Q_H. Note that Q_H can include the same h multiple times, whereas Q_I and Q_R consist of a list of unique pages p . Beside this, the application orchestrates several sub-processes:

- A *URL input handler* P_{input} that takes the next URL from Q_I and adds it into its corresponding host h.
- A *URL output handler* P_{output} that selects a URL from Q_H to be crawled, based on the targeting function and puts it into Q_R
- A *bandit-based host handler* P_{bandit} that selects the next h based on a given function and inserts it into Q_H.
- An *online classifier* $P_{classifier}$ that classifies new URLs based on a given set of parameters and the target function.[42]

Figure 3.2 illustrates the flow throughout our approach. The crawling process starts with a number of initial seed pages (0), which are fed into Q_I. Then, P_{input} pulls the first page p from Q_I. Before adding p into the corresponding h, the page is classified by $P_{classifier}$. In the online setting, $P_{classifier}$ starts off with an empty model as no training data (pages) are available so far. Whenever $|H| \neq 0$ and $\exists h \in H : \left|R_h^t\right| > 0$, P_{bandit} selects one host h based on the given $s(h)$ and λ (1). The selected h is inserted into Q_H and hosts in Q_H are processed by P_{output}. For each host, the URL with the highest confidence for the target class is selected and pushed into Q_R (2). The reordered pages are now ready to be handled by other components of the crawler. After downloading (3) and parsing (4) the page, the newly found links are added into Q_I (5). In addition, the label of the crawled pages is returned as feedback to $P_{classifier}$ which updates its classification model (6).

This component is fully distributed in nature, which in practice means that processes operate independently and some of them work faster than others. We optimized all the underlying processes in order to maximize the system throughput, this is, to minimize the probability that Q_R gets empty and the crawler has to wait for new pages. Additionally, we implemented a mechanism to delay the process P_{bandit} whenever the crawler is busy, as it might occur a slight delay in receiving the feedback for the action a_t, when the system calculates the score for a_{t+1}.

[41] By filtering we mean the removal of duplicate and unwanted pages (like certain file extensions like videos, images, etc.).

[42] We use the *MOA Java library* 2012.08 from `http://moa.cms.waikato.ac.nz` introduced by [Bifet et al., 2010].

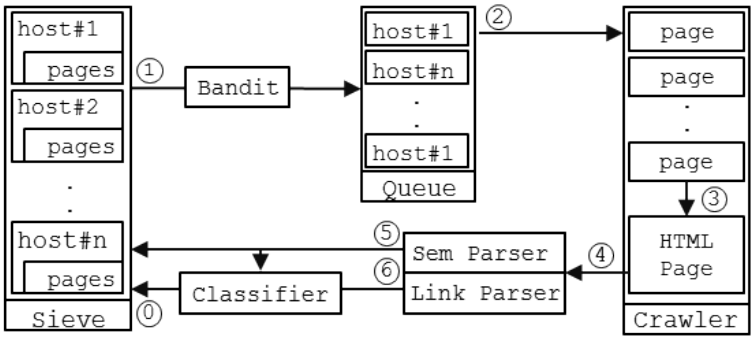


Figure 3.2: The architecture of the focused crawler for semantic annotations.

3.3.2 Research Data

We employ, similar to the related work, a static dataset for our experiments in order to isolate ourselves from changes in the content of a web page and the underlying hyperlink graph, as well as other factors such as the availability of web page hosts.

In particular, we make use of the web graph datasets, which we extracted in our former work, analyzing the overall structure of the Web [Meusel et al., 2014c].[43] The graph dataset was derived from the 2012 web corpus provided by the Common Crawl Foundation. From this original corpus we derive a subset of around 5.5 million web pages which are reachable from one root URL. This was randomly selected from the URLs retrieved by crawling the pages of the *Open Directory Project*.[44] The dataset includes 455 848 different hosts.

In the following, using *Apache Any23* which allows the programmatic detection of semantic annotations in HTML pages (compare Section 2.3), we marked all web pages within the subset which deploy semantic annotations containing (a) at least one statement and (b) more than four statements using Microdata.

From (a) we acquired 1.5 million pages, which comprise 27.4% of the whole 5.5 million sub-dataset. From (b) we acquired 179 383 pages, which is 3.25% of the whole 5.5 million sub-dataset.

With the subset and the structured information retrieved in (a), we will run most of the experiments to evaluate our approach for the general task of gathering efficient semantic annotations from the Web. With the subset and the structured information retrieved in (b), we will run a secondary experiment and show that our approach is adaptable to different objectives in the area of semantic annotations crawling.

Furthermore, we extracted from the whole corpus an optimization dataset (c). Web pages from this dataset are not contained in (a) or (b). The dataset (c) consists of 100 000 pages from over 1 000 different hosts with a balanced distribution of

[43] The dataset is available on the web site of the Web Data Commons project: `http://webdatacommons.org/hyperlinkgraph`.

[44] `http://dmoz.org`

labels (same number of web page deploying semantic annotations and web page without any semantic annotations). We use the dataset to find an optimal feature and parameter setup for our experiments.

3.3.3 Experiments Description

In the following, we describe the sets of different experiments which were performed in order to evaluate focused crawling for semantic annotations.

Feature and Parameter Optimization

As mentioned in Section 3.2, the selection of features and classification algorithm has a major influence on the final web page selection performance. Therefore, in an upfront experiment using the dataset c, we determined the most adequate combination of features, classifier and parameter configuration (number of hashes, classifier dependent settings, etc.). We randomly select one page after the other, first letting the classifier predict the label and then training it with the real label.[45] We repeat this process ten times for each different configuration and measured both the overall accuracy of the classifier and the running time needed for classification and training of the whole dataset. The time needed to train and classify becomes important in case the crawler is able to gather the next page but the classifier has not finished the prediction process.

We experiment with two different online classification algorithms, namely Naive Bayes (which was also used by [Chakrabarti et al., 2002b]) and Hoeffding Trees [Zliobaite et al., 2011]. Furthermore, we experimented with features obtained from the URL of the unseen page, features gathered from the parent pages and features from the sibling pages and all possible combinations.

We engineer those features using a variety of configurations, for instance filtering by token length and replacing number tokens by the constant string *[NUMBER]*. Finally, we evaluated the performance of the classifiers with the tokens hashed into different number of features, ranging from $5k$ to $20k$.

Crawling for General Semantic Annotations

Using the optimal configuration, our first series of experiments aims to validate that our approach can effectively steer a crawling process toward web pages that embed any kind of semantic annotations (Dataset a).

In the first step, we compare our approach to a standard BFS approach and the typical approach to building focused crawlers for specific topics, i.e., using a static classifier. We run our implementation on the described dataset with a static classifier which we initially trained with $100k$, $250k$ and $1\,000k$ pages, and in comparison we run several crawls that incorporate online classifiers. In a second step, we determine

[45] This reflects the real operating mode of a crawler, except that a real crawler might have some delay in when feedback is available.

which scoring function for the hosts leads to the highest number of crawled pages that are relevant to our target function. We run several crawls incorporating different scoring functions for the bandit-selection, turning off the greedy component of the algorithm ($\lambda = 0$). Likewise, we selected the page with the highest confidence score from the bandit-chosen host. In a next step, we try different static values for λ to report the influence of the randomness for the best performing scoring functions. Additionally, we will show the effect of a decaying λ_t using different decaying factors m.

Crawling for more Specific Semantic Annotations

In a second series of experiments, we change the objective for our crawling function. Therefore, we have defined relevant pages as those that embed semantic annotations within its HTML code regardless of its kind and quantity. We now narrow down further this definition in order to measure the adaptability of our techniques to different objective functions. We want to reward only pages that embed at least five statements using the semantic markup language Microdata (Dataset b). Pages using Microdata typically use the schema.org vocabulary and provide more complex descriptions of the information present in the page. The number of statements is a rough quality criteria in that we filter out pages that provide only minimal detail. As an example, a HTML page describing movies that contains at least five statements might include the facts that:

1. this page describes a movie,
2. the described movie has the title *Se7en*,
3. the described movie has a rating of 8.7,
4. the described movie was published in 1995 and
5. this information was maintained by *imdb.com*.

Note although the presented example contains useful information following the restriction mentioned before, it might also happen that a page contains the same useless information multiple times, which would also trigger our defined objective function.

Runtime Experiments

Finally, we analyze the runtime of the different scoring functions. This is an important consideration because crawling is essentially a matter of resources, and it might happen that the crawler requires an unacceptably large time budget in order to select a new page being crawled. We are aware that this consideration depends on the crawling strategy and the implemented policies, which have been optimized consciously.

For the experiments, using dataset (a) and dataset (b) we always used the same initial seeds. In addition, as a large number of our experiments depends on sampling – especially those that test different bandit functions – we repeated each experiments up to five times and report the average.

3.3.4 Evaluation Metrics

The main objective of focused crawlers is to maximize the number of relevant pages gathered while minimizing the number of not relevant pages which are downloaded and parsed during the crawl. In order to evaluate the effectiveness of our approach, we use a *precision* measure that reports on the ratio of retrieved relevant pages to the total number of pages crawled. A page is considered to be relevant when it supports the objective crawling function, this is, whether the page deploy semantic annotations or not at all.

3.4 Results

In the following, the results of the experiments described in Section 3.3.3 are presented. First the most adequate configuration for the online classification algorithm is determined using dataset (c). Then, the results for the general objective function based on dataset (a) are described. In the last part of this section, the object function is narrowed down using dataset (b). The curves in the drawings within this section are calculated using the smoothing spline method.

3.4.1 Online Classification Optimization

Using the optimization dataset (c), we tested all possible combinations of feature sets, token manipulation (e.g., filtering short tokens), classification algorithms, and number of hashes. Based on the results we find that:

- Hoeffding Trees perform overall slightly better in comparison to Naive Bayes (81% accuracy versus 77%).
- Hoeffding Trees need up to 10-times more time in comparison to Naive Bayes.
- Ignoring tokens shorter than three characters and a replacement of numbers by a constant string works best.
- Using $10\,000$ hashes produces the best results.
- Information from parent URLs, whenever available help to improve the performance.
- Adding sibling information into the set of features downgrades the performance consistently.

Due to the fact, that the time for learning and predicting is crucial and we cannot effort to let the crawler wait for the classifier to finish its prediction we selected the Naive Bayes classification algorithm for the subsequent experiments. The remaining settings are based on the findings presented above.

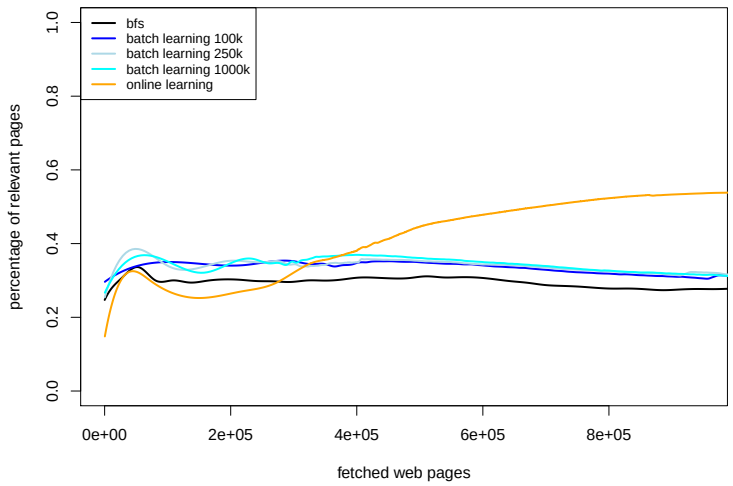


Figure 3.3: Percentage of relevant fetched pages during crawling comparing batch-wise and online Naive Bayes classification.

3.4.2 Offline versus Online Classification

Static classification has been a dominant method for focused crawling. Our first set of experiments compared the performance of batch with online learning in our domain of interest. We ran different crawls using pre-trained classification models learned on a subset of $100k$, $250k$ and $1\,000k$ randomly selected pages. Figure 3.3 shows the number of relevant retrieved pages of static approaches (blue lines). The orange lines show the ratio of relevant pages gathered by a crawler equipped with an adaptive online model which was trained completely from scratch during the crawling process. In addition, we include the data series (black line) representing a pure breadth-first search approach (BFS).

The performance numbers of static-based classification are slightly higher than the number of BFS. Remarkably, online learning is able to increase notably the amount of relevant pages crawled after $400k$ fetches. At the end of the crawl, the adaptive approach is able to collect $539k$ relevant pages whereas the best static one (trained with $250k$ examples) fails to collect $200k$ of those. This trend is similar with Hoeffding Trees, although the difference in performance diminishes when the model is trained with $1\,000k$ pages. Still, we note that there is a decreasing performance rate for static classification approaches. The online learner also underperforms on the first half of the crawl. This is because the model is empty at the beginning and needs to be trained in subsequent iterations, where static models have a slight edge due to their knowledge advantage.

Figure 3.4 reports the accuracy over time of the classifiers present in Figure 3.3. The x-axis shows the number of fetched web pages, whereas the y-axis describes

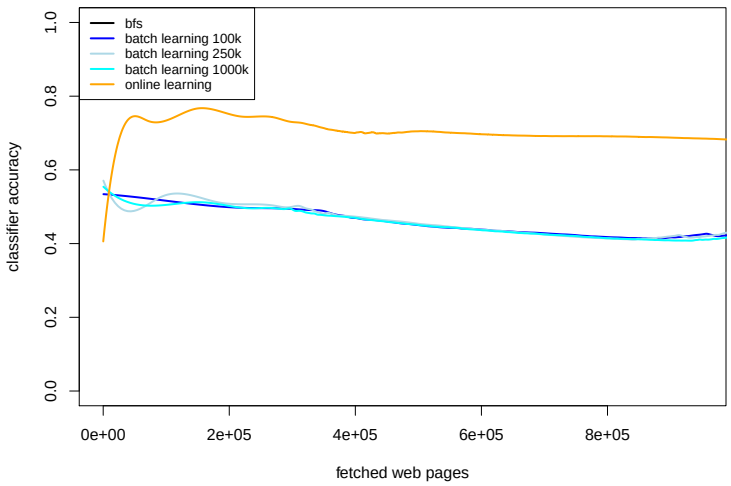


Figure 3.4: Development of the accuracy of the classification model of batch-wise and online Naive Bayes learning during crawling.

the ratio of correctly classified to crawled pages. The saturated accuracy of the static classification approaches ranges between 0.55 and 0.45 where the adaptive model reaches 0.7 in the long run. The rather poor performance of the static classification models can be explained by over-fitting on the training dataset.

3.4.3 Evaluation of Different Bandit Functions

We now look into the interplay of bandit algorithms and the different functions to calculate the expected value of a host h (presented in Section 3.2). This first analysis will not include any randomness ($\lambda = 0$), to observe the real impact of the different setups, and compare them against a random selection, a BFS approach and a pure online classification based selection (*best score function*).

Figure 3.5 shows the percentage of relevant pages retrieved during the crawling of one million pages. Firstly, all tested functions lead to higher number of retrieved relevant pages than the a pure random selection (black line) or a BFS (grey line). Furthermore, except for the Thompson Sampling based selection (TS) all the scoring functions outperform online classification on its own (and therefore using static classification approaches). The highest performance rate is achieved by the *success rate function*, which simply measures the ratio between relevant and non-relevant pages for a host. Here we are able to fetch around $628k$ relevant pages out of one million. The three combinations of *best score functions* with (a) TS, (b) *absolute good* and (c) *success rate* yield the second best results. Regarding the TS-based functions, we see a sharp increase in relevant pages retrieved at early stages of the crawl. This decreases toward the end of the measurement series ending

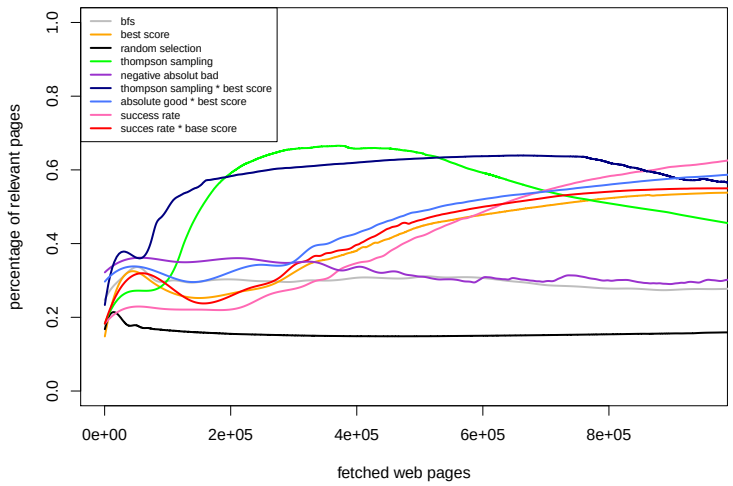


Figure 3.5: Percentage of relevant fetched pages during crawling comparing different bandit functions ($\lambda = 0$).

up gathering between $550k$ and $600k$ relevant pages. In comparison, the other mentioned functions present a positive trend toward the end of the series.

Having identified the best performing scoring functions, we now want to focus on the *explore/exploit* problem. We run the best performing bandit-based selection functions, namely *absolute good* in combination with the *best score*, the *success rate* and the combination of *success rate* and *best score* and measure the impact of different values λ. We tested the named functions using different fixed values of λ (we report on the best ones) and compared to the corresponding gathering rate without any random selection. We did not consider the TS approach, as it already includes an element of randomness through sampling from a beta-distribution [Chapelle and Li, 2011].

Figure 3.6 shows the impact of different λ values for our three selected functions. We can state that the use of a random factor in the cases of the *best score* and the *absolute good function* fails to increase the number of crawled relevant pages. Regarding the functions that include *best score*, using a fixed λ greater than zero reduces the number of relevant pages. The above result may suggest that λ greater than zero may not be beneficial. Figure 3.7 zooms into the first $400k$ crawled pages and shows that there is a positive impact of including a random factor $\lambda > 0$, lifting the relevant page rate from 0.3 to 0.4. However, this effect diminishes when the amount of crawled pages reaches $1\,000k$.

The above results support our initial intuition that a decaying lambda may provide the best results overall. We compare the performance of linear decaying functions for λ (described in Section 3.2), with a fixed $m = 10k$ (value learned on

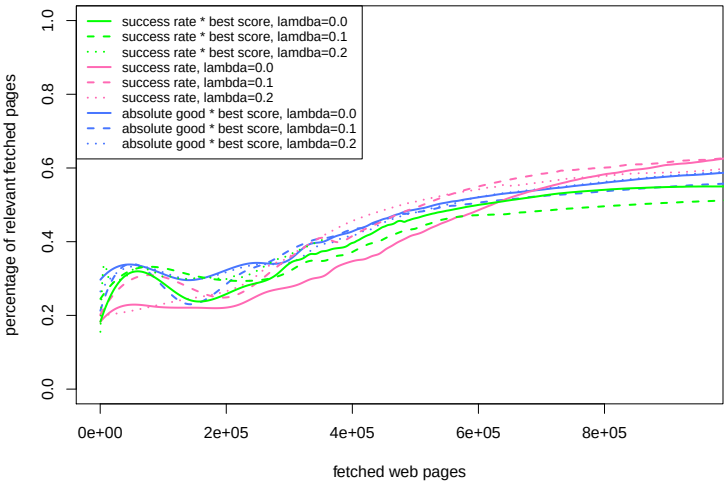


Figure 3.6: Percentage of relevant fetched pages during crawling pages comparing best performing bandit functions with different λ values.

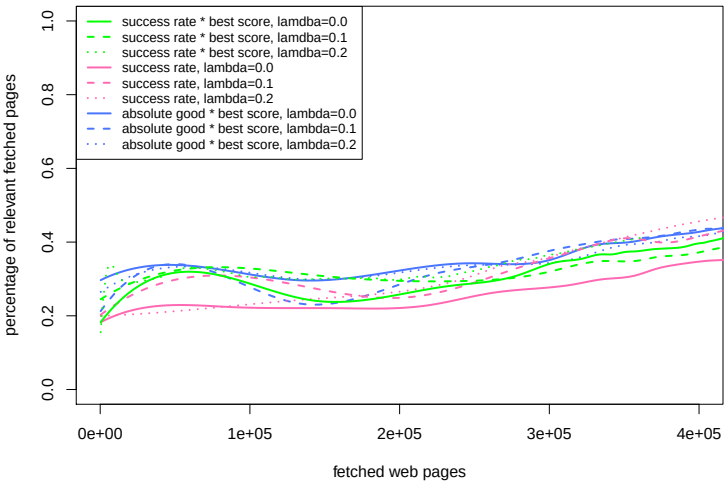


Figure 3.7: Percentage of relevant fetched pages during crawling of first $400k$ pages comparing best performing bandit functions with different λ values.

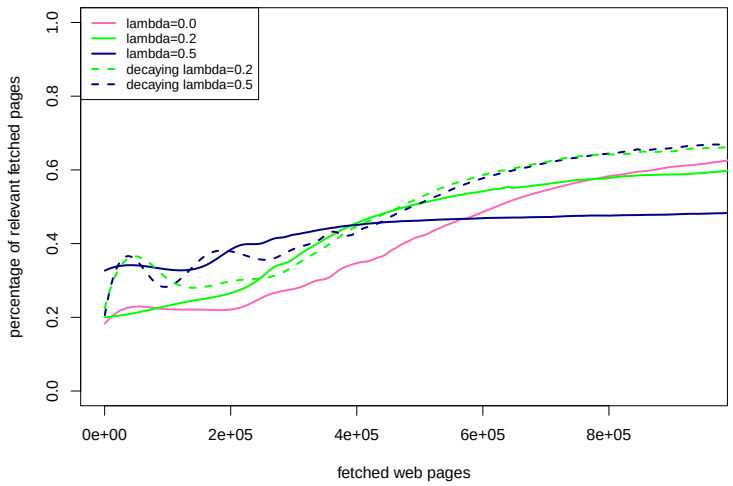


Figure 3.8: Percentage of relevant fetched pages during crawling comparing success functions with decaying and static λ.

an independent development set). Figure 3.8 shows the number of crawled relevant pages of the *success rate function* for the static and decaying λs. In addition to the already used $\lambda = 0.2$, we also show the results of a larger $\lambda = 0.5$ in order to increase the randomness and potentially learn more in the earlier stages of the crawl. Results show a positive impact of a decaying λ for the percentage of fetched relevant pages, achieving the maximum amount of relevant pages ($673k$). The positive effect is especially noticeable with $\lambda = 0.5$ – with no decaying factor, one out of two page selections are random (yielding the worst results), however when the decaying factor comes into play this negative effect disappears in the long run.

The results in this section are summarized in Table 3.1. The results show a 10% improvement of the best performing method for online classification (Naive Bayes) over the best performing result for static classification (HoeffdingTree with $1\,000k$ training set) and a 26% improvement of the best combined bandit-based approach (*success rate* with decaying $\lambda = 0.5$) on top of online classification alone.

3.4.4 Adaptability to More Specific Semantic Annotations Crawling Tasks

In this experiment, we change the focus of our crawler and reward only pages with at least five statements embedded using the semantic markup language Microdata.

We compare the BFS approach and the best score function with the best performing configuration from the former section: (1) *absolute good* · *best score* $\lambda = 0.0$, (2) *success rate* · *best score* $\lambda = 0.1$, (3) *success rate* $\lambda = 0.2$ and (4) *success rate* with decaying $\lambda_t = 0.5$ and $m = 10k$. Figure 3.9 shows the percentages of fetched

Table 3.1: Overview of percentage of crawled relevant pages after one million crawled pages.

Selection Strategy	Percentage of Relevant Pages
Random	.159
BFS	.291
Naive Bayes ($100k$ Training Set)	.312
Naive Bayes ($250k$ Training Set)	.316
Naive Bayes ($1\,000k$ Training Set)	.311
Naive Bayes (Online)	.534
HoeffdingTree ($100k$ Training Set)	.408
HoeffdingTree ($250k$ Training Set)	.381
HoeffdingTree ($1\,000k$ Training Set)	.482
HoeffdingTree (Online)	.512
Thompson Sampling ($\lambda = 0.0$)	.452
Thompson Sampling · Best Score ($\lambda = 0.0$)	.562
Negative Absolute Bad ($\lambda = 0.0$)	.300
Absolute Good · Best Score ($\lambda = .0$)	.589
Success Rate ($\lambda = 0.0$)	.628
Success Rate · Best Score ($\lambda = 0.0$)	.550
Success Rate ($\lambda = 0.1$)	.628
Success Rate ($\lambda = 0.2$)	.600
Absolute Good · Best Score ($\lambda = 0.1$)	.558
Absolute Good · Best Score ($\lambda = 0.2$)	.590
Success Rate (decaying $\lambda_t = 0.2$)	.662
Success Rate (decaying $\lambda_t = 0.5$)	.673

relevant pages for the first one million crawled pages. From the figure we perceive that all tested functions perform remarkably better than the BFS approach. The overall achieved rates are around five times smaller than the rates we reached for the more general objective function. However, for this objective function the amount of relevant pages among all the ones in the crawl is around eight times lower (0.04 vs. 0.27). In addition, after crawling one million pages, the bandit functions also outperform the online classification based selection strategy. Like in the previous experiment using a success rate based function tend to gather the highest number of relevant pages, with the *success rate function* with $\lambda = 0.2$ reaching a percentage of relevant crawled pages of 0.12 in the first million crawled pages. In comparison, online classification based selection ends up with a ratio of 0.08. Finally, in this experiment a decaying λ performed comparably to using a fixed λ value.

3.4.5 Evaluation of Runtime

In the previous experiments we have shown that the combination of online classification and a bandit-based approach leads to a higher percentage of relevant crawled pages for both tested objectives. We now assess what is the processing overhead incurred by our classification approaches and the current implementation for page

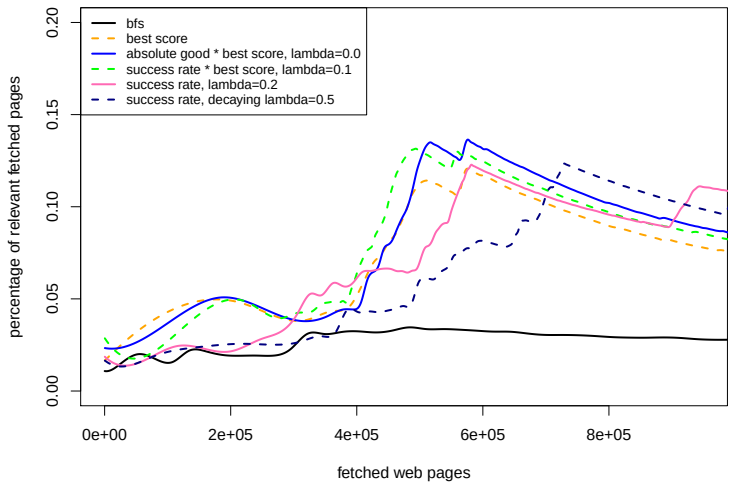


Figure 3.9: Percentage of relevant fetched pages during crawling aiming for pages with at least five Microdata statements.

selection. This time is critical, as when comparing to a BFS approach, we cannot venture to drop below the average processing time to crawl and parse a page as in this case the crawler process would have to wait for our selection.

The theoretical time which is needed to select one page for crawling is mainly influenced by four factors:

- The **number of different hosts**, as the bandit needs to go through all of them on each iteration.
- The **runtime of the scoring function** for the hosts.
- The **selection of the final page** from the selected host, which depends on the **number of pages per crawl** that are ready to crawl.
- The **time to add the feedback** to the system, which includes the time needed to (re-)train the classifier and update internal scores.

In terms of a random selection the runtime for the scoring function and the time to add the feedback to the system are omitted.

Figure 3.10 shows the average time in milliseconds for the bandit approaches presented before to determine the next page to be crawled. To make results comparable, we also include the fully random selection approach. We can observe, that scoring functions not making use of the Thompson Sampling, where internally a beta-distribution needs to be calculated perform better than a pure random selection. The average time to selection one page range below 50ms for the dataset we used in our experiments. The two functions, making use of a beta-distribution need up to 300ms to select one page. Looking deeper into these functions, we noticed that the creation of the beta-functions and the selection of the random value needs over 75%

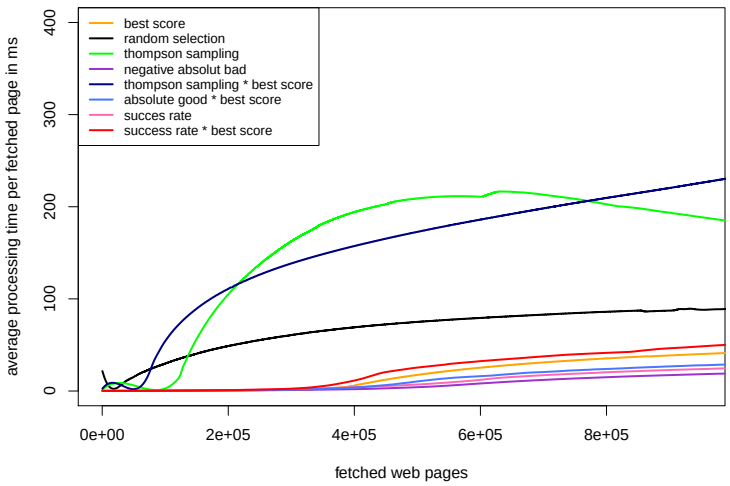


Figure 3.10: Average processing time to select one page over time.

of the whole processing time. In order to estimate the overhead of including our selection policies into a fully-fledged system one needs full measurements of the standard crawling cycle: establishing a connection, downloading the page, parsing and extracting new links. Taking a broad general estimate from an existing BFS crawler *Ubi-Crawler* [Boldi et al., 2004], which needs $800ms$ to fully process one page per thread, our *selection* policy would incur in an overhead of less than 10%, as we need not more that $50ms$ for page selection. In comparison to this, we would boost the percentage of relevant crawled pages by factor three.

3.5 Conclusion

Adapting state-of-the-art mechanisms and extending them with a bandit-based approach to overcome the *explore/exploit* problem, we have introduced a focused crawling approach targeting web pages deploying semantic annotations. The current implementation, which is publicly available, is designed to replace the *selection policy* of existing crawlers. Furthermore, an integration into *Apache Nutch* is available as download as well. We have shown that the use of online classification, in comparison to static classifiers, can achieve better results in this domain being able to collect over 10% higher numbers of relevant pages for a given objective function. Furthermore, our results show that grouping pages based on their host and making use of features shared by this group empowers the selection strategy for pages and improves considerably the resulting percentage of relevant crawled pages. We demonstrated that a bandit-based selection strategy, in combination with a decaying learning rate (decaying λ) overcomes the *explore/exploit* problem during

the crawling process. Our results show that it is possible to increase the percentage of relevant crawled pages in comparison to a pure online classification-based approach by 26% (compare Table 3.1).

Narrowing the focus of our crawler to web pages using Microdata (where we can extract at least 5 statements) we have shown that our approach can gather 66% more relevant pages within the first million than a pure online classification based approach. In general, estimating the expected value of an host using the *success rate function* in combination with always selecting the pages with the highest confidence for the target class tends to lead to the best results. Going beyond *precision* considerations, we have analyzed the runtime performance needed by the current implementation of our approach to select relevant pages for crawling and showed that we need, in average, 50ms to select a new page. The results presented in this chapter demonstrate that a focused crawler using a bandit-based selection with online classification is capable of effectively gathering semantic annotations.

An open question, which needs to be further analyzed is to which extend the approach can be applied to even more specific target functions. In particular, is the strategy also able to discover web pages deploying semantic annotations which are only contained in less than 0.1%, or even less than 0.01% of the whole corpus. In such a case, the classification algorithm as well as the bandit-based scoring function would not retrieve examples of web pages supporting the objective function too frequently.

In addition a direction of further research is the estimation of the optimal decaying factor, as we have shown that depending on the objective function a higher or smaller factor is better. Furthermore, also a dynamic factor is imaginable, which attaches itself to the number of hosts where too less information are available.

Chapter 4

Data Extraction from Web Corpora

In the previous chapter, we have shown that the use of focused crawling for the purpose of collecting web pages deploying semantic annotations is promising. Although employing such an approach increases the harvesting rate of desired web pages significantly, crawling in general is still challenging and can be costly [Cambazoglu and Baeza-Yates, 2015].

An alternative approach is the extraction of semantic annotations directly from large web corpora. Thanks to the efforts of initiatives like the *Common Crawl Foundation* and the *Lemur Project*[46] such web corpora are available to public. Before the efforts of these initiatives only companies like the search engine companies could effort the maintenance of large web corpora, which they did not make publicly available.

Although the use of such public web corpora eludes most crawling-specific challenges, such as seed selection and the effort of detecting duplicated URLs, the sheer size of the data embodies a challenges which needs to be faced. To process such large corpora in a reasonable timespan, applications need to execute the extraction process in parallel. As the potential of horizontal scaling is in most cases limited by the number of processors (cores) and the amount of random-access-memory (RAM) of one machine, parallelization over different machines (also known as vertical scaling) is necessary. Vertical scaling implies the need of several servers, or server clusters. Especially for smaller and/or non-profit organizations like Universities, the purchase of larger amounts of servers is not feasible, particularly as those servers might only be used for a short timespan. A more and more popular alternative to the own purchase of computing units is the usage of on-demand servers provided by so-called *cloud services*. Providers of cloud infrastructures like Amazon Web Services (AWS)[47] or Microsoft Azure[48] provide, whenever needed,

[46] `http://lemurproject.org/`
[47] `http://aws.amazon.com/`
[48] `http://azure.microsoft.com/`

servers in various sizes and quantities. In order to provide a high flexibility for any kind of use cases, the cloud providers separated the provided components, such as servers for computations, possibilities for storage, and the transfer between the different components. Also each component has its own pricing model. Therefore, applications making use of such services needs to be tailored towards the available components in order to effectively (not only in resources but also based on financial aspects) make use of the available services.

Within this chapter, we present a framework which is designed to efficiently parse web pages contained in large web corpora. In particular, we adapt the framework to extract semantic annotations from billions of web pages provided by the Common Crawl Foundation. The described framework is horizontally and vertically scalable and can be executed within the cloud infrastructure of AWS[49]. Before explaining the details of the application in Section 4.2, Section 4.1 briefly introduces the different available web corpora. In Section 4.3 we describe the adaptions made to the framework and show case its usability for the extraction of semantic annotations by harvesting semantic annotations from billions of web pages contained within three different corpora of the Common Crawl Foundation. Section 4.4 visualizes the adaptability of the framework to other use cases by outlining concrete projects making use of the framework. The chapter concludes in the final section.

The initial concept of the framework's integration into AWS was prototyped by Hannes Mühleisen, who also applied this early version to a reduced subset of the 2012 corpus provided by CC [Mühleisen and Bizer, 2012]. The description of the workflow of the actual framework has already been published in [Seitner et al., 2016].

4.1 Public Web Corpora

In the following, we list the most important providers of publicly available web corpora and briefly discuss the provided data for the purpose of extracting semantic annotations. We are aware that there might be other web corpora, but to the best of our knowledge the here mentioned are the largest and most comprehensive ones.

The Lemur Project

Up to now, the Lemur Project has released two large web corpora. The *ClueWeb09* crawl[50], gathered in 2009 contains over one billion web pages in ten different languages with a compressed size of over 5 TB. Due to the extraction date the data is rather outdated, as most of the semantic markup languages were just introduced after 2009. An additional important note is that when regarding the underlying

[49] We have chosen the services provided by AWS due to two reasons. First, at the beginning of our research AWS was the most comprehensive cloud provider. Second, AWS granted our research by an education grant which allowed us the usage of their services for free, within a certain frame.

[50] `http://www.lemurproject.org/clueweb09.php/`

hyperlink graph of the pages contained in the document, it was found that the LSCC is only 3% of all pages and that a large number of pages is disconnected. Therefore it is questionable to which extend the corpus is representable for the Web with respect to the findings of [Broder et al., 2000].

The *ClueWeb12* crawl[51], released concurrently with the writing of this thesis, contains over 700 million web pages, which were collected between February and May 2012. The corpus, which is available since the beginning of 2013 focus primary on the English part of the the Web. This restriction influences the representativity and coverage of the corpus and hence limits usability of this corpus for the purpose of analyzing the deployment of semantic annotations in the Web.

Common Crawl Foundation

In 2012, the *Common Crawl Foundation* started publishing crawl corpora of the Web. Their first three corpora, containing documents gathered in 2008/2009, 2009/2010, and the first half of 2012 were collected using a customized version of the Apache Nutch crawling framework [Khare et al., 2004]. To gather those corpora, CC initialized their crawler with a ranked list of seed page from the previous crawls and discovered new pages in a *breadth-first search* fashion. The resulting collection of web documents is highly interlinked by hyperlinks from pages to others, as shown in our former work [Meusel et al., 2014c, Meusel et al., 2015c, Lehmberg et al., 2014a]. Especially the corpora containing over 3.83 documents gathered in the first half of 2012 was at this time the largest public available web corpora. [Spiegler, 2013] analyzed this corpus in particular and reported among others that around 55% of the pages originate from com-top-level domains, and includes large amounts of sites from the video portal *Youtube* and the blog publishing service *blogspot*. In December 2012 CC announced the cooperation with the search engine company *blekko* [Lindahl, 2012]. Starting from this point in time, CC switched their crawling strategy and primary *re-crawled* the page index of *blekko*. Unfortunately, it is not known how this index of 6 billion URLs is exactly created but a PageRank-like technique is used. Analyzing the resulting hyperlink graph from those web crawls underlines this switch in the used *selection strategy*. [Meusel et al., 2014d] found that in the corpus of April 2014 still 91% of the nodes (pages) are connected, but that the SCC only contains of 19% of the pages. We also found that the IN component contains almost 50% of all pages which underlines the large number of *isolated* entry points of the crawler. Till the purchase of *blekko* by *IBM* in the beginning of 2015 CC published 19 crawl corpora using this strategy. In May 2016, CC replaced the information given by blekko with a seed list of 400 million URLs from the SEO consulting company *Moz*[52]. Each corpora, provided by CC, is split into multiple *WARC* files[53] containing a set of independent web pages. Although the corpora are

[51] `http://www.lemurproject.org/clueweb12.php/`

[52] `https://moz.com/`

[53] WARC files following the ISO 28500:2009 standard: `http://www.iso.org/iso/catalogue_detail.htm?csnumber=44717`.

created with a bias towards popular web pages, CC did not restrict their crawl to a particular language or geographical region, which causes the crawls to be more representative for the whole Web than the one provided by the Lemur Project.

AltaVista

For the sake of completeness, we also mention the AltaVista web page connectivity dataset, distributed by Yahoo! as part of the *WebScope program*[54]. The datasets contains over 1.4 billion nodes which are highly disconnected: half of the nodes are isolated (no links incoming or outgoing) and the largest strongly connected component is less than 4% of the whole graph, which makes it entirely unrepresentative. Furthermore, we have no knowledge of the crawling process, and URLs have been anonymised, so no investigation of the causes of these problems is possible. In addition, it is only the graph without the content of the documents, which makes it impossible to extract semantic annotations.

4.2 Overall Extraction Workflow

In this section, we first present the workflow of the general extraction framework which is scalable and can be executed using the services provided by AWS. The subsequent section discusses the application-specific adaptations which were used to extract semantic annotations embedded in web pages using Microformats, RDFa, and Microdata.

The framework is written in Java and tailored towards the AWS environment, other than the work presented by [Bugiotti et al., 2012]. The frameworks in general aims to enable a painless and efficient parsing of large document collections, in this case the web corpora of CC. This workflow is described in Figure 4.1. In general, (1) a queue (SQS) is filled with the references to all files which need to be processed. Then (2) a number of servers is requested and is started automatically, which perform all the same process on all available cores: (A) ask the queue for the next file, then (B) download the file to the server and (C) process the file. After finishing (D) the output is written back to the storage (in the default case S3) and (E) the queue is notified that the file is parsed and can be removed from the queue. After the queue is empty, (3) the results can be collected from S3. In this process only three actions needs to be triggered manually via a command line interface, the remaining actions, including the communication between the different components in the cloud environment are done automatically.

The current version of the framework[55] is configured to parse web pages contained in WARC files using the official *International Internet Preservation Consortium* (IIPC) Java web archive library[56].

[54] `http://webscope.sandbox.yahoo.com/catalog.php?datatype=g`

[55] Version 1.0.3. which is available here: `https://www.assembla.com/spaces/commondata/subversion/source/HEAD/WDCFramework/tags/1/0/3`.

[56] `https://github.com/iipc/webarchive-commons`

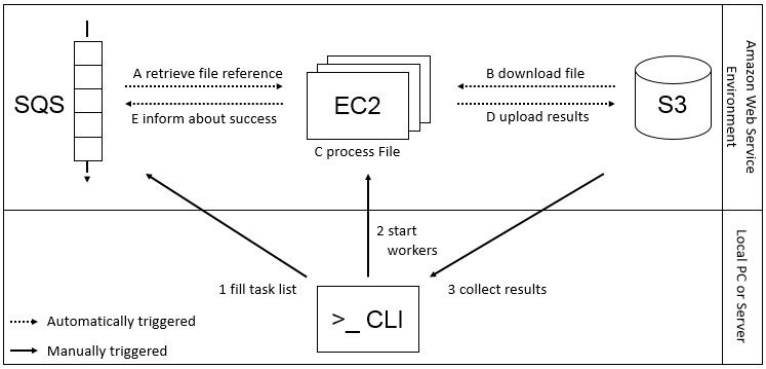


Figure 4.1: Overview of the web corpus extraction framework workflow.

4.3 Extraction of Microformats, RDFa, and Microdata

In order to customize the framework for specific extractions like the extraction of semantic annotations by Microformats, RDFa, and Microdata, step C needs to be adapted (compare Figure 4.1). In this section, we present the implementation which was necessary to adapt the framework to the task of extraction semantic annotations and also present the outcome of the utilization of this customized version on three corpora of CC.

As already described in Section 2.3, we make use of Any23 to parse the HTML code of a web page and extract semantic annotations. Some of the Any23 extractors need to internally build up the whole HTML document as document object model (DOM) to extract the markup language-specific semantic annotations. This can be time and resource consuming. Hence, we implemented a relaxed semantic markup language detector based on regular expressions. This detector upfront checks if the HTML contains a set of markup language-specific terms. Table 4.1 lists the used terms for each of the different markup formats. In case of non-existence of any of those terms in the HTML code, we reject the HTML page and do not try to parse the semantic annotations.

We used the before described setup to extract semantic annotations from three corpora of CC. Table 4.2 summarizes the basic statistics, including size, number of different web sites and web pages of the corpora from the years 2012, 2013, and 2014. All three crawl corpora contain more than two billion HTML pages which originate, depending on the crawl from at least 12 million different web sites. In addition, the table shows the sizes of the extracted data, which are also publicly available on the web site of the Web Data Commons project[57]. The number of quads refers to the number of RDF statements which could be extracted (compare Section 2.3).

[57] http://webdatacommons.org/structureddata/index.html

Table 4.1: List of regular expressions used for relaxed markup language detection.

Markup Language	Regular Expression
RDFa	`(property|typeof|about|resource)\s*=`
Microdata	`(itemscope|itemprop)\s*=`
Microformats geo	`class\s*=\s*("|')[^"']*geo`
Microformats species	`class\s*=\s*("|')[^"']*species`
Microformats xfn	`<a[^>]*rel\s*=\s*("|')[^"']*(contact|acquaintance|friend|met|co-worker|colleague|co-resident|neighbor|child|parent|sibling|spouse|kin|muse|crush|date|sweetheart|me)`
Microformats hcalendar	`(vcalendar|vevent)`
Microformats hcard	`vcard`
Microformats hlisting	`hlisting`
Microformats hresume	`hresume`
Microformats hreview	`hreview`
Microformats recipe	`hrecipe`

Table 4.2: Overview of extracted semantic annotations from three selected Common Crawl web corpora.

Time Period of Data Collection	Jan-Jun 2012	Winter 2013	Dec 2014
Common Crawl Corpus Statistics			
# Pages in Crawl	3 005 629 093	2 224 829 946	2 014 175 679
# Web Sites in Crawl	40 600 000	12 831 509	15 668 667
Corpus Size (compressed in TB)	40	44	46
Semantic Annotations Corpus Statistics			
# Quads	7 350 953 995	17 241 313 916	20 484 755 485
# Pages with Semantic Annotations	369 254 196	585 792 337	620 151 400
# Web Sites with Semantic Annotations	2 286 277	1 779 935	2 722 425
Corpus Size (compressed in GB)	101	332	373

For each of the three extractions, we made use of around 100 *AWS spot instances*[58] of type $c1.xlarge$. Exemplary, using the mentioned setup, we were able to parse the CC 2014 corpus, consisting of 46 TB compressed data, within less than 24 hours, while spending less than US$300.

4.4 Additional Use Cases

During the design and implementation of the framework, the aspect of adaptability for other extraction use cases was dominant. Ensuring a high adaptability of the framework allows other researches to perform their own extraction by writing some lines of additional code, exploiting the scalability of the framework.

In the following, we present four additional research projects which made use of the framework presented before, in chronological order.

Hyperlink Graph Extraction In 2014, we used the framework in order to extract the hyperlink graph from two CC corpora, as already mentioned in Section3.1. The

[58] Spot instances enable the bidding on spare Amazon computing capacities. More information can be found on the web page `https://aws.amazon.com/ec2/spot/`.

used methods parsed the original HTML content of the page and extracted all kinds of hyperlinks from the content and saved the pairs of URLs (URL of the page and the URL of the discovered link) back to the storage system of AWS. Using those pairs, we extracted the so far largest public available hyperlink graph [Meusel et al., 2015c, Meusel et al., 2014c, Lehmberg et al., 2014a]. Unfortunately, due to the design of the framework, we could not use it directly to filter all extracted URLs and remove only those URLs which we also had crawled pages in the corpus. We further discuss this limitation in the subsequent section.

Web Table Extraction Another project by [Lehmberg et al., 2014b] targeted on the extract of HTML tables from web pages. The project was inspired by the work of [Cafarella et al., 2008] and made use of the corpora provided by the CC and the framework used in this thesis. The extracted tables contains somehow structured information, but without a well-defined meaning. In order to detect the tables as well as removing of so called layout tables, which are HTML tables solely used to make the page look nice, the authors integrated classification models together with several rules into the framework.

Hyperlink Anchor Tag Extraction Similar to the extraction of the hyperlink graph from a web corpus, [Bryl et al., 2015] used the framework in order to extract surface forms for entities of the *Wikipedia* encyclopedia. They adapted the framework in the way, that whenever a web page contained a link to a Wikipedia web page, they extracted this particular link together with the anchor texts from the HTML page. Using additional sources together with various ranking mechanisms, they came up with a set of surface forms for Wikipedia entities (e.g., *BRD* for *Germany*).

Hypernymy Extraction One of the most recent work [Seitner et al., 2016], used the framework in order to extract hypernymy relations from the textual elements of HTML pages. Using customized patterns, similar to *Hearst patterns* the authors generated a very large database, which provides a frequency based ranked list of hypernymy for an input term. Especially in the area of natural language processing, such knowledge bases based on web data are useful.

4.5 Discussion and Conclusion

Making use of the scalable framework, we were able to collect semantic annotations from the large web corpora provided by the Common Crawl Foundation. In particular we have extracted semantic annotations from over 7 billion web pages contained in three different corpora within together less than 3 days, spending less than US$1 000.

The benefit of using public web corpora, besides the fast and cheap extraction is the omission of the multiple challenges of crawling. In addition, as the underlying

web corpora are publicly accessible, others can reproduce the results and compare their approaches and methods.

The disadvantage of using a web corpora is the dependence on the provider. For most subsequent investigations based on web corpora, detailed knowledge about the curation process is necessary, hence a high level of transparency on the side of the provider is required. Especially information about the used seeds as well as the implemented *selection strategy* are essential in order to interpret the outcome of studies correctly. Furthermore, web corpora might be outdated and therefore might not reflect the current characteristics of the Web.

Limitations and Alternative Approaches The major limitation of the current version of the framework becomes visible for the use case of extracting the hyperlink graph from the web corpus. For this use case, the framework is capable of extracting the hyperlinks from the HTML pages, but it cannot index the graph directly. This is due to the absence of inter-thread communication within the framework, where all threads work independently in parallel. Therefore, for some use cases, task-specific applications for further post-processing are required, which we also implemented for the use cases presented in Section 4.4.

A possible alternative, which overcomes the mentioned drawback, is the execution environment *Hadoop* [White, 2012]. In contrast to our custom-made framework, Hadoop enables not only the parallel execution of applications over various different servers but also the communication of the servers among themselves. This communication requires a coordination node (in the context of Hadoop called *master*), which manages all the messages and data passed between the different execution threads. In 2012, when starting the development of the framework, the overhead of the *master* of the Hadoop version, indicated not to use Hadoop for the purpose of extracting semantic annotations from web pages. This observation might not be valid any more at this point in time, as further improvements of Hadoop and related applications like *Spark*[59] have been going on.

Besides the direct improvements of Hadoop in the last years, addition application packages like *Apache Pig* have become available. Those applications in general make use of the distributing nature of the Hadoop infrastructure and allow for certain use cases an optimized processing. Meaning, that in some cases a holistic end-to-end application might not be the best solution and a step-wise approach, as used in the mentioned use cases might be more applicable.

[59] `http://spark.apache.org/`

Chapter 5

Comparison of the Extraction Approaches

In the two previous chapters of the thesis, we have presented and evaluated the potential of two different strategies to collect semantic annotations from the Web. Both approaches focused crawling (compare Chapter 3) and the extraction from public web corpora (compare Chapter 4 show promising potential for the extraction of semantic annotations.

Although the usefulness of both strategies for the task has been evaluated it is due to the size of the Web and its dynamic almost impossible to collect all relevant web pages, as already described in Section 3.1.1. Therefore, no matter which strategy is used the resulting collection of semantic annotations will always be only a sample.

In order to select the most appropriate sample for the purpose of profiling of the deployment of semantic annotations in the Web, this chapter first discusses the aspect of *representativity* of samples retrieved by the two proposed strategies. As the discussion shows, although the rate of obtained web pages deploying semantic annotations is higher by using focused crawling the resulting corpus is less representative with respect to the whole Web, as hose provided by the Common Crawl Foundation. Having determined the most suitable source of samples for the purpose of the thesis, Section 5.2 discusses the problems of errors arising from the usage of samples. In particular the section calculate the highest expected sampling error for the three extracted corpora. The last section concludes this chapter underlining that the use of semantic annotations extracted from the CC web corpora is the most appropriate chose with respect to the focus of the thesis.

5.1 Representativity

An essential concept referred to as *representativity* needs to be considered whenever any kind of statement is obtained from a sample and should also hold for the overall population. This kind of setup is commonly used especially in science. Here,

whenever the data is too large to be included in the methodology as whole, a sample is drawn which is used in place of the whole population. Doing this implies that the sample is *representative* for the population, but in most cases this representativity is not discussed in any further detailed. This might not be a problem, whenever a sufficient large random sample is used. But in some circumstances random sampling might not be applicable or might not lead to a sample which still represents the major characteristics of the overall population which are essential for a specific application. Therefore, creating a representative sample is highly application dependent [Kruskal and Mosteller, 1979a]. In addition, as the population, especially the Web changes steadily over time, the detected distributions might diversify as well. This effect is discussed in [Gummer, 2015] in more detail.

An example are power-law-like distributions, which can be find in large graphs such as hyperlink graphs derived from the Web. Although a solid number of works exist which analyze characteristics of the Web based on samples, almost none discuss the representativity of the used sample. The work by [Leskovec and Faloutsos, 2006] is an exception and focuses directly on the question which methods can be used to generate a representative sample from large graph. As based on observations by [Broder et al., 2000] and [Clauset et al., 2009] in the Web power-law-like distributions can be found almost everywhere, therefore specific sampling methods are needed so that the conclusions drawn from the sample are also valid in the original graph – the Web. The authors find, that although random node selection is a comparable good sampling method, *Forest Fire* is more sufficient in order to preserve graph specific characteristics, such as the degree distribution. Unfortunately the authors restrict their-selves to the sample of large graphs, but do not consider the problem of creating those large graphs from real populations. This makes it difficult to draw any benefits from their findings for the estimation about the representativity of a web corpus for the whole Web.

Within the field of social science, which besides others uses polls and surveys to gain further knowledge, the representativity of the sample with respect to the overall population is controversially discussed. [Schnell et al., 2011] claims that the concept of representativity is fuzzy and not useful at all. The authors argue that only a random sample is representative and that no single measure exist that can be used to measure how representative a non-randomly drawn sample is. But the authors also state that over the boarders of their field other definitions are used which can be mainly summarized by the different meanings stated by Kruskal and Mosteller. In their work [Kruskal and Mosteller, 1979a], [Kruskal and Mosteller, 1979b], and [Kruskal and Mosteller, 1979c], the authors summarize the different meanings of the term *representative sampling* within different application domains.

They identify in general six different meanings for representative sampling which are most commonly used [Kruskal and Mosteller, 1979b]:

1. General acclaim for data.
2. Absence of selective forces.
3. Miniature of the population.
4. Typical or ideal case(s).
5. Coverage of the population.
6. Vague term, to be made precise.

Especially in the field of statistics, the authors further identified three additional meanings for representative sampling [Kruskal and Mosteller, 1979c]:

1. as a specific sampling method.
2. as permitting good estimations.
3. as good enough for a particular purpose.

In an ideal case, we want the data which serves as object of investigation to be gathered with the absence of a selective force and to be a miniature of the population, which covers the whole population.[60] In order to achieve this ideal state, we would need to randomly select pages and sites from the whole Web and include them into our corpus. This would require an upfront knowledge of web pages and web sites which are contained in the Web. Further, we would require a random access to every of those web pages. Both requirements are, based on the discussion in Section 3.1, not realistic due to the dynamics of the Web. In theory, it would be possible to randomly generate URLs, which would be time consuming as it can be reckon that a large number of those URLs do not lead to a HTML page or all to the same page.[61] In addition the distribution of different characters within URLs is not totally random, which in the end would create a sample which is again not random.

The sampling of the Web is aggravated by the lack of knowledge how the Web really looks like. Even search engine companies like Google are not able to index the whole Web. In order to collect the, for their application relevant part of the Web they deploy selection strategies which are based on the general idea of PageRank and therefore implement the idea of popularity. Meaning that they introduce a selective force to create their sample.

Unfortunately, this selected force can be found within both evaluated extraction techniques. In the case of the usage of existing web corpora, we know that they are also collected based on a popularity measure, similar to the one used by the search engine companies. Under the premise that most visitors of the Web to some extend rely on the search results returned by a search engine and therefore stick more or less to the Web which is presented by this entry point, we could argue that the web

[60] Of course, all the other first six meanings are somehow applicable but the mentioned three might be the most obvious one.

[61] Depending on the configuration of the hosting web application server, request for non-existing URLs of a web site could be redirected to the landing page of this web site.

corpora provided by CC are somehow *good enough* for the purpose of representing the *known* or *public* Web. This factor is especially important for the analyzing of the overall deployment of semantic annotations in the Web and its development over time.

In the case of using focused crawling, the strategy favors web pages embedding semantic annotations over those who do not. This selective force, with respect to the goal of profiling the deployment of semantic annotations in the Web, contradicts the paradigm of an absence of a selective force for creating a representative sample of the Web. For other investigations conducted in the following chapters of the thesis focusing especially on a particular dataspace, semantic annotations collected by this strategies can be used, as long as no implications are reflected to the overall Web. Especially, as potentially more web pages deploying semantic annotations can be collected for a given resource limit by this strategy.

As this thesis does not only focus on the analysis of the dataspace stretched by semantic annotations in isolation, but tries to answer questions about its overall deployment in the whole Web, we make in the following chapters use of the semantic annotations collection from the CC corpora (compare Chapter 4).

5.2 Sampling Errors

Although we have discussed that the public web corpora provided by CC are appropriate for the goal of this thesis, all analyses in the following chapters are performed based on samples of the Web, which size is referred to by N. Although the size of our sample n is comparable large in contrast to samples used for example in *social science*, it is necessary to measure the *standard error* for our observation $\overline{y}$ introduced through sampling as discussed in [Horvitz and Thompson, 1952]. The standard error $S(\overline{y})$ is defined in Equation 5.1, where $V(\overline{y})$ is the variance of the observation.

$$S(\overline{y}) = \sqrt{V(\overline{y})} \tag{5.1}$$

In order to interpret the standard error the *confidence interval* c is used, stating the estimated maximal error percentage for a given confidence level. c is the product of the variance and the z-value for a given confidence level (compare Equation 5.2).

$$c = z^* S(\overline{y}) \tag{5.2}$$

The calculation of the confidence interval allows the estimation of the maximal expected error of an observation in the sample for a certain confidence level, e.g., 95% or 99%.

For observed percentages ($\overline{y} = \overline{p} \quad \text{with} \quad \overline{p} \in [0, 1]$) within a sample, representing a small fraction of the population ($f = \frac{n}{N}$), the variance $V(\overline{p})$ can be estimated by the Equation 5.3 as described in [Lohr, 1999].

Table 5.1: Confidence intervals of the percentage of web pages and web sites deploying semantic annotations for confidence levels 0.95 and 0.99 for the extractions of 2012, 2013, and 2014.

Extraction	Overall #	with Semantic Annotations #	%	Confidence Interval 0.95	0.99
Web Pages					
2012	3 005 629 093	369 254 196	0.1229	0.00001	0.00002
2013	2 224 829 946	585 792 337	0.2633	0.00002	0.00003
2014	2 014 175 679	620 151 400	0.3079	0.00002	0.00003
Web Sites					
2012	40 600 000	2 286 277	0.0563	0.00007	0.00009
2013	12 831 509	1 779 935	0.1387	0.00019	0.00025
2014	15 668 667	2 722 425	0.1737	0.00019	0.00025

$$V(\overline{p}) = \frac{\overline{p}(1-\overline{p})}{n} \tag{5.3}$$

The equation calculates the variance only based on the observation and the size of the sample, without taking the size of the population into account. This means that the variance of an observation stays unchanged, no matter if the sample fraction is 0.1 or 0.01. Such an estimation is in particular useful - and used - whenever the population is really large or infinite, or the sample is very small (e.g., for election polls).

In a first step, we follow the premise that the number of HTML pages is almost infinite (as already discussed within Section 3.1 and the same holds for the number of web sites in the Web. We can use the formula (compare Equation refeq:varestimation) to estimate the confidence interval for the observations of the percentage of web pages and web sites deploying semantic annotations as listed in Table 4.2.

Table 5.1 shows those intervals for the percentage of web pages and web pages in the crawl (sample) which make use of at least one of the described markup languages. Even if we apply a confidence level of 99% the maximal error for these observations is 0.02%.

In a second step, in order to get a feeling about the range of the confidence intervals for different observations, we calculate the confidence level for the two fictional observations 0.5 (50%) and 0.01 (1%).[62] Even for the smallest sample (2013), in terms of the number of web sites, the error ranges between 0.036% and 0.007%, respectively for a confidence level of 99%. The assumption about the infinite number of HTML pages is reasonable but the number of different web sites is at least finite. Therefore, the requirement for a small f might not hold in case of web sites. In [Horvitz and Thompson, 1952] this problem is discussed and the authors introduce a penalizing factor within the estimation of $V(\overline{p})$. Equation 5.4

[62] Based on the Equation 5.3 observations larger than 0.5 do not need to be considered, as the parts of the dividend can be flipped.

Table 5.2: Refined confidence intervals of the percentage of web sites deploying semantic annotations for confidence levels 0.95 and 0.99 for the extractions of 2012, 2013, and 2014.

	# Web Sites		Sample	Web Sites with Semantic Annotations		Confidence Interval	
Extraction	in the Web	in the Crawl	Fraction f	#	%	0.95	0.99
2012	233 000 000	40 600 000	0.1742	2 286 277	0.0563	0.00006	0.00008
2013	271 000 000	12 831 509	0.0473	1 779 935	0.1387	0.00018	0.00024
2014	288 000 000	15 668 667	0.0544	2 722 425	0.1737	0.00018	0.00024

for a refined variance $V'(\overline{p})$ is derived from this idea and incorporates the fact that a larger sample produces more precise estimations.

$$V'(\overline{p}) = (1-f)V(\overline{p}) = (1-\frac{n}{N})\frac{\overline{p}(1-\overline{p})}{n} \tag{5.4}$$

In order to calculate f, we derived the number of web sites in the Web N, which were registered during the time the samples were taken from the domain name industry briefs of *Verisign*.[63] Making use of those estimations about the size of the total population, we can use the formula of Equation 5.4 and refine the confidence intervals for observations on web site level. We find that the confidence intervals decrease but are still comparable to the intervals before, as shown in Table 5.2.

Again for the smallest sample fraction (2013), we calculate the ranges of the confidence intervals for 0.5 and 0.01 and result that the expected error ranges between 0.035% and 0.007%, respectively for a confidence level of 99%.

5.3 Conclusion

In the previous sections, we have analyzed the two alternative strategies to obtain semantic annotations from web pages using Microformats, RDFa, and Microdata directly or from web corpora. As a representative sample of the Web is needed, we have shown that the use of corpora gathered by general crawls is the most adequate solution with respect to the goal of the thesis. We further have shown, that based on the size of the crawl the sampling error for a confidence level of 99% is expected to be between 0.035% and 0.007%, which we think is sufficiently small enough to rely on our results.

[63] In particular we use the reports for July 2012 (`https://www.verisign.com/assets/domain-name-brief-july2012.pdf`), and the fours quarter reports of 2013 (`https://www.verisign.com/assets/domain-name-report-april2014.pdf`) and 2014 (`https://www.verisign.com/assets/domain-name-report-march2015.pdf`). The reported numbers are the numbers of registered domains, which does not necessary mean that any HTML page is available for these domains.

Part II

Analysis of Semantic Annotations

Chapter 6

Overall Adoption of Semantic Annotations

The former part of this thesis has focused on approaches to collection semantic annotations from the Web. In this and the following chapters, different profiling related aspects of the dataspace of semantic annotations are empirically analyzed.

In order to generate a holistic profile for the dataspace, consecutive studies are carried out. In particular, this section covers the area of the adoption of semantic annotations in the Web. Knowledge about the adoption is useful in order to get a first estimation about the size and the spread of the dataspace. Analyzing the topics covered by semantic annotations embedded in web pages using Microformats, RDFa, and Microdata, allows us to discover markup language-specific differences and to generate a coarse-grained distinction of the entities described by semantic annotations. These results are essential for further studies, focusing on the utility of semantic annotations for certain use cases. In addition, an analysis of the changes of the adoption over time allows the discovery of trends of future development.

The chapter starts with an introduction to the field of profiling data or dataspace, according to the profiling dimensions described in [Naumann, 2014]. The introduction explains the level relevance of the different dimensions for to purpose of profiling the dataspace of semantic annotations in HTML pages. Having introduced profiling in general and discussed the relevant dimensions, Section 6.2 presents the results of the empiric analyze of the adoption and topical coverage of the semantic annotations based on the three CC corpora of 2012, 2013, and 2014. Due to the chronological order of the corpora, the section also examines the changes of the two profiling aspects over time. Section 6.3 presents related work in this research area with a specific focus on profiling of web data. The final section concludes the findings.

6.1 Introduction to Data Profiling

The section introduces the different dimensions of data profiling using the classification schema described in [Naumann, 2014]. The classification schema is derived from classic profiling of relational data, as found in databases, and the profiling of data from different sources, e.g., to support a data integration task. Within its second part, this section explains what dimensions of the profiling classification schema are covered for the purpose of analyzing the adoption of semantic annotations in the Web. Furthermore, it is explained why other dimensions are not considered or not directly applicable for the dataspace of semantic annotations.

6.1.1 Different Dimensions of Profiling

Being rooted in the database community, data profiling deals with methods for analyzing and describing datasets. Based on the work of [Naumann, 2014] profiling-related tasks (in particular of databases) can be categorized in different dimensions as shown in Figure 6.1. The categorization of the tasks which arise in data profiling, depends primarily on the number of sources, the dataset is retrieved from. Profiling datasets from only one source is further separated in the analysis of single columns or multiple columns. In the field of the analysis of single columns classic problems like the *cardinality* and *data type* detection are mentioned. The analysis of *dependencies* - functional, conditional and approximated - is a separated task, affecting multiple columns.

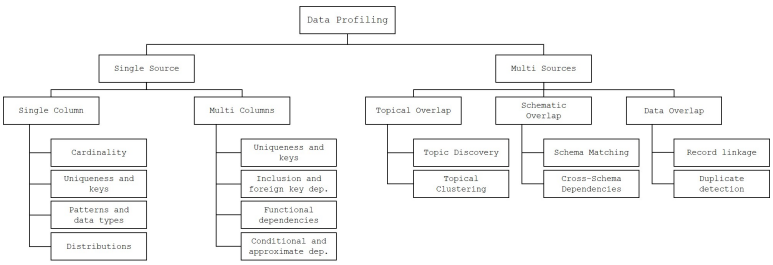


Figure 6.1: Hierarchical description of the different data profiling dimensions, adapted from [Naumann, 2014].

Besides the profiling of datasets retrieved from one single source, profiling of data from multiple sources becomes more and more interesting. Especially for the task of *data integration*, where information from multiple data providers are integrated in order to generate a single consistent dataset, the profiling of the overlap of the different datasets is essential [Doan et al., 2012].

Due to Naumann, this profiling of the overlap can be categorized in three different parts:

Topical Overlap: This part of the profiling aims to discover the general topics, based on a use case, which are covered by the data from the different datasets.

Schematic Overlap: As it cannot be assumed that all the different sources of data use the same schema to describe similar entities, this area of profiling focus on the matching of the different schema as well as the identification of dependencies across all available schema.

Data Overlap: Besides the topical and schematic overlap, the data overlap covers the profiling of the number of entries from different datasets describing the same object. In addition, it also aims to identify the number of records which contain duplicated information.

As mentioned also by Naumann, the task of profiling is in most cases driven by a subsequent task. Such task can be rather general like *data cleansing*, but also use case-specific like the integration of the data into a prize comparison platform. Some dimensions of the profiling hence might be more interesting and important than others, depending on the use case.

6.1.2 Profiling Semantic Annotations in HTML pages

The thesis focuses on the profiling of the dataspace of semantic annotations in HTML pages. Those semantic annotations are provided by millions of data providers. Where in a first place, we do not have any knowledge about the used semantic markup language as well as the used vocabulary. Furthermore, the topics covered by the semantic annotations are unknown. Consequently, the profiling of each individual source of semantic annotations does not reveal general insights about this dataspace. Therefore we omit the profiling of each data source in separate and restrict ourselves to the dimensions of profiling dedicated to "multi sources" [Naumann, 2014]. Furthermore, in order to generate a use case-independent profile, the examination of keys in general and the calculation of dependencies are not useful in a first place. Besides, as later studies show, semantic annotations rarely contain identifiers.

Especially in this chapter, we mainly focus on the dimension of *topical overlap* besides the analysis of the overall adoption of semantic annotations in the Web. The two dimensions are not only discussed for one point in time, but for multiple points in time, allowing also the analyzes of changes and the detection of trends in the deployment of semantic annotations.

We restrict the dataspace to semantic annotations embedded by Microformats, RDFa, and Microdata in HTML pages together with commonly used vocabularies. Therefore, the profiling of the *schematic overlap* is not within the focus of our work. We are aware, that there are some data providers which employ their own vocabulary, which makes a deeper analysis of the schematic overlap necessary. But within the scope of the thesis, we will dismiss such rare cases.

In addition to the dimensions proposed by [Naumann, 2014], we carry out studies focusing on the compliance of semantic annotations towards their vocabulary definition. This dimension of profiling is not mentioned in the classification schema by Naumann, as it might not be a common case in the area of relational databases. The results are presented together with a study about the *data overlap* in Chapter 7.

Extending the study of changes over time, Chapter 8 focuses on the aspect of changes in the schema within a certain time frame. Using a data-driven approach, we analyze how the changes in the official definition and changes in the actual adoption influence each other. This aspect is also not covered by the classification schema of [Naumann, 2014] as relational databases normally implement their own schema, and do not align to a global, unique schema.

6.2 Profiling of the Adoption of Semantic Annotations

Having discussed the various dimensions of profiling and their applicability for the dataspace of semantic annotations, in this section we first present a study of the adoption of semantic annotations in the Web. Further this section provides insights of the adoption of the three different semantic markup languages in the Web. In addition, the topical coverage of the semantic markup language-specific semantic annotations is examined. All those aspects are also analyzed with respect to their evolution over time. Upfront, we briefly describe the corpora which are used and the measures which are applied to carry out our studies.

6.2.1 Research Data and Measures

In order to perform an analysis of the deployment of the dataspace of semantic annotations in the Web, we make use of the data extracted from the web corpora provided by the Common Crawl Foundation. An overview of the three different corpora, extracted from crawls performed in 2012, 2013 and 2014, can be found in Table 4.2.

As recap to Chapter 5, although we have shown that the corpora by CC represent the popular part of the Web, they include also, due to the employed *selection strategy* a bias. As CC used *PageRank* to collect web pages, for popular web sites more web pages are contained in the crawl than for other web sites. We can observe the consequences in the corpora of 2013 and 2014. Although the corpus of 2013 contains more web pages, the one of 2014 contains web pages from a larger number of different web sites. To elude this bias in following studies, we measure the adoption of semantic annotations, certain semantic markup languages or classes based on the percentage of web sites embedding them in the HTML code of their web pages, with respect to the overall number of web sites contained in the corresponding crawl. We report the deployment for all semantic markup languages together as well as separated for all three corpora. From the changes of the percental deployment

within each corpus, we can conclude the trend for the deployment of the different semantic markup languages.

Furthermore, we use a similar approach to profile the topics contained in the dataspace. The selection of a vocabulary and a class in order to annotate the information contained within a web page are an indicator for the topics described. We measure the percentage of web sites making use of particular class and calculate a topical distribution for all three corpora.

Besides the analyses on web sites level, we could repeat the same studies on web page level. But the results would be almost meaningless with respect to the overall deployment in the Web because the number of web pages in the Web is theoretically infinite (as discussed in Section 3.1) and the corpora contain more web pages of popular web sites.

6.2.2 Overall Adoption

In a first step, we analyze the overall number of web sites embedding semantic annotations in general and in particular for the different semantic markup languages. As mentioned before, the numbers and percentages are only reported on web site-level.

From Table 4.2 we see that the overall percentage of web sites within the popular part of the Web embedding semantic annotations increased over the last years from 5.6% in 2012 to over 25% in end of 2014.[64] Further, we analyze the adoption of semantic annotations by each semantic markup languages in separation based on their percental deployment based on the number of all web sites embedding semantic annotations (compare Table 6.1). The second column for each year shows the relative deployment based on the number of all web sites making use of at least one markup language by at least one of the crawled pages. In addition, the tables state the relative change between 2012 and 2014.[65]

Starting in 2012, we can observe that the MF`hCard` is the most commonly used semantic markup language (66%) which also does not change till 2014 (40%), although the relative amount decreases by almost 40%. The second most deployed semantic markup language in 2012 is RDFa with around 22%, followed by the MF`XFN`. Microdata is not very common in 2012 where it is only used by around 6% of the web sites embedding semantic annotations. Moving to 2013 and 2014, we see that the deployment of Microdata increases and Microdata becomes more and more popular. In 2014, the relative deployment of Microdata has increased by factor four in comparison to 2012. Also we see an increase in the deployment of RDFa from 2012 to 2013, overall the grows diminishes till 2014 and the percental usage of RDFa is back at around 20%. We observe the strongest decrease of deployed semantic annotations for the semantic markup languages MF`hResume`, MF`XFN`, and MF`geo`.

[64] As discussed in Section 5.2 the observed distributions might differ by less than 0.035% for a confidence level of 99%.

[65] $\Delta_{12,14}$ represents the relative change between the year 2012 and 2014.

Table 6.1: Number and percentage of web sites deploying semantic annotations in the years 2012, 2013 and 2014, divided by Microformats, RDFa, and Microdata.

	2012		2013		2014		
	#	%	#	%	#	%	$\Delta_{12,14}$
RDFa	519 379	.227	471 406	.265	571 581	.210	−0.076
Microdata	140 312	.061	463 539	.260	819 990	.301	+3.908
MF`geo`	48 415	.021	23 044	.013	20 261	.007	−0.649
MF`hCalendar`	37 620	.016	20 981	.012	24 208	.009	−0.460
MF`hCard`	1 511 855	.661	995 258	.559	1 095 517	.402	−0.391
MF`hListing`	4 030	.002	2 584	.001	3 167	.001	−0.340
MF`hRecipe`	3 281	.001	3 530	.002	3 476	.001	−0.110
MF`hResume`	1 257	.001	262	.000	155	.000	−0.896
MF`hReview`	20 781	.009	12 880	.007	13 772	.005	−0.443
MF`species`	91	.000	109	.000	96	.000	−0.114
MF`XFN`	490 286	.214	195 663	.110	170 202	.063	−0.708

6.2.3 Microformats Adoption

Based on its conceptual design, the topics described semantic annotations embedded by Microformats are related to their Microformats-specific markup language (and not on the vocabulary used), e.g., MF`hRecipe` to describe cooking recipes and MF`hCard` to describe persons or organizations.[66] Therefore, we can directly retrieve the topical distribution from Table 6.1. Most web sites making use of Microformats, describe persons (MF`hCard`) and their relations (MF`XFN`). Besides, we find events (MF`hCalendar`) and geographic locations (MF`geo`) described. In addition, but similar to the other topics with a decreasing evolutionary trend, we find reviews (MF`hReviews`).

Among other reasons, the strong adoption of MF`hCard` and MF`XFN` can be affiliated to the integration of Microformats into content management systems (CMS) and blogging systems like *wordpress*[67] and *blogspot*[68]. Wordpress offers a large variation of free layout templates which already annotate the content of the CMS with the respective Microformats classes.[69]

6.2.4 RDFa Adoption

Within the most recent corpus (2014), we find over 571 thousand web sites embedding semantic annotations using RDFa. This is 21% of all web sites embedding semantic annotations. In comparison to the former extractions (2012 and 2013), we can observe a slight decrease.

In contrast to Microformats, RDFa (as well as Microdata) can be combined with any vocabulary in order to define information within HTML. Table 6.2 outlines the

[66] Please see Table 2.1 for a topical overview of the different Microformats.

[67] `http://wordpress.com`

[68] `http://blogger.com`

[69] An overview of the variety of available templates implementing at least Microformats and their number of active installations can be found on this web page: `https://de.wordpress.org/plugins/tags/microformats`.

most frequently used vocabulary-namespaces in combination with RDFa in 2014.[70] We find `og`, `ogp` as well as `fb2008` to be most frequently used, which all belong to the OGP vocabulary. Furthermore, we find the vocabularies `dc`, `dv` and `foaf`, each being deployed by more than 10% of the web sites making use of RDFa. Overall the distribution of vocabularies is strongly dominated by vocabularies promoted by Facebook.

Table 6.2: Most common used vocabularies together with RDFa in 2014.

	Web Sites	
Vocabulary (Namespace)	#	%
`og`	210 014	.367
`ogp`	139 457	.244
`fb2008`	99 666	.174
`foaf`	69 927	.122
`dc`	66 060	.116
`dv`	57 101	.100
`sioc`	48 740	.085
`ogp/fb`	46 822	.082
`skos`	12 736	.022
`cc`	12 145	.021

In the 2014 corpus, we find over 1.9 million different classes and 22 thousand different properties deployed by web pages making use of RDFa. Based on observations of former studies [Meusel et al., 2014b], we have already found that a large fraction of classes and properties are either typos or web site-specific and are only embedded by one web sites. Also in the 2014 corpora this can be observed as we only found 908 different classes and 2 358 different properties which are deployed by at least two different web sites.

In order to gain insights into the topics described by semantic annotations embedded in HTML pages using RDFa, we more closely inspect the most frequently deployed classes, based on the relative number of web sites. With respect to the changes over time, we notice that most of the 20 most frequently used classes do not change significantly between the three corpora. Table 6.3 lists those most commonly deployed classes with their total number of web sites embedding them and the percental adoption based on the total number of RDFa web sites within the corresponding corpora. Seven out of those twenty classes belong to the Open Graph Protocol vocabulary (prefix: `og`) and are in particular used to create a connection between entities on a web page to the Facebook ecosystem.[71] Besides the general annotations used for web sites (website, document, image, breadcrumb), we find two major topics described by RDFa: e-commerce (products, reviews, companies) and blogging (blog, blogposts, comments).

[70] For an explanation of the namespaces please compare Section 2.2.

[71] Please note that those classes appear without any namespace. For reasons of readability, we added the `og` prefix.

Table 6.3: Number of web sites making use of RDFa and a specific class in 2012, 2013, and 2014 ordered by their deployment in 2014.

	Class	2012 #	2012 %	2013 #	2013 %	2014 #	2014 %	$\Delta_{12,14}$
1	`og:website`	56 573	.109	71 590	.152	164 324	.287	+1.639
2	`og:article`	183 046	.352	167 554	.355	141 679	.248	−0.297
3	`foaf:Image`	44 644	.086	46 505	.099	53 467	.094	+0.088
4	`foaf:Document`	49 252	.095	45 542	.097	51 694	.090	−0.046
5	`dv:Breadcrumb`	9 054	.017	39 561	.084	49 771	.087	+3.995
6	`sioc:Item`	33 141	.064	29 521	.063	33 019	.058	−0.095
7	`og:blog`	58 971	.114	29 629	.063	27 913	.049	−0.570
8	`og:product`	19 107	.037	13 813	.029	14 592	.026	−0.306
9	`skos:Concept`	13 477	.026	11 873	.025	12 600	.022	−0.150
10	`sioc:UserAccount`	19 331	.037	12 632	.027	12 217	.021	−0.426
11	`dv:Review-aggregated`	6 236	.012	5 266	.011	5 945	.010	−0.134
12	`sioc:Post`	6 994	.013	2 703	.006	4 642	.008	−0.397
13	`dv:Rating`	4 139	.008	3 603	.008	4 331	.008	−0.049
14	`og:object`	4	.000	63	.000	3 133	.005	+710.716
15	`og:activity`	3 303	.006	2 037	.004	2 757	.005	−0.242
16	`og:company`	6 758	.013	3 105	.007	2 644	.005	−0.644
17	`sioc:Comment`	3 339	.006	2 639	.006	2 438	.004	−0.337
18	`sioc:BlogPosting`	3 936	.008	2 639	.006	2 431	.004	−0.439
19	`gr:Offering`	1 342	.003	2 199	.005	2 196	.004	+0.487
20	`vcard:Address`	3 167	.006	2 225	.005	2 173	.004	−0.377

We further calculated the number of different properties which are used by each web site using RDFa. We find that on average around five properties (5.3) are used.

Facebook Ecosystem Integration As already briefly mentioned in Chapter 2, Facebook requires web sites to annotate their semantic annotations with four mandatory properties. The `title`, the `type`, e.g., a video, a `url`, as well as an `image` for objects is necessary to be integrated into the social network. Table 6.4 outlines the most frequently used properties by web sites making use of at least one of the two different OGP namespaces. The percentage reflects the part of web sites making use of a certain property. We can see that for the older OGP namespace `ogo`, over 90% of the web sites make use of three of the four mandatory properties, where only 78% deploy `ogo:image`. For web sites using the `ogm` namespace its slightly less, but still over 75% of the web sites use the mandatory properties. Other properties than the one in the list are hardly used. This indicates a strong influence of dominant data consumers on the adoption of semantic annotations, which we analyze in Chapter 8 in more detail.

6.2.5 Microdata Adoption

As we have seen in the former analysis of the most common used semantic markup languages, Microdata is the most trending one. Similar to RDFa, Microdata can be used together with any vocabulary but based on our observations its mainly used together with two different vocabularies. In the corpus of 2014, we find the schema.org vocabulary being used by 84% of all web sites making use of Microdata.

Table 6.4: List of most common deployed properties of the OGP vocabulary in 2014, separated by the two different namespaces.

Property	% Web Sites	Property	% Web Sites
ogo:title	.974	ogm:title	.919
ogo:url	.965	ogm:url	.809
ogo:site_name	.936	ogm:description	.770
ogo:type	.925	ogm:type	.765
ogo:image	.786	ogm:image	.763
ogo:description	.618	ogm:site_name	.731

The second most commonly used vocabulary is *data-vocabulary*, the predecessor of schema.org, which is used by 14% of all Microdata web sites. The remaining 2% of web sites use other vocabularies like `foaf` or `dcterms`. The absence of other vocabularies within the Microdata dataspace is already noticeable in the corpus of 2012. In the 2012 corpus, the distribution of the two main vocabularies is slightly shifted and more web sites make use of data-vocabulary in that year, which is explicable as the first release of schema.org (before release `0.91`) was published in June 2011.

In the corpus of 2014, we observe over 22 thousand different classes and over 315 thousand different properties. Again, we have a look at those numbers for classes and properties which are deployed by at least two different web sites. Similar to the observations for RDFa annotations, we only find conspicuous less diversity. In particular 2 998 different classes and 30 788 different properties are deployed by more than one web site.

Table 6.5 depicts the 20 most commonly deployed classes using Microdata in 2014 and their adoption (total and relative) in the two years before. As mentioned before, the schema.org vocabulary is mostly dominate. This is also reflected in the top classes, where we only find five classes belonging to the `dv` vocabulary. Topical-wise, we find semantic annotations about blogs and news embedded with Microdata (`s:Blog`, `s:Article`, `s:BlogPosting`) and e-commerce related information (`s:Product`, `s:Offer`, `dv:Product`, `dv:Offer`). Besides, we find information about organizations and persons (`s:Organization`, `s:LocalBusiness`, `s:Person`, `dv:Organization`). Furthermore, for most of the mentioned topical domains above, we also find ratings and reviews (`s:Rating`, `s:AggregateRating`, `dv:Review-aggregate`).

Regarding the number of different properties deployed on average by one web site, we found that semantic annotations embedded in HTML pages using Microdata are described on average by a rather small number of entities in comparison to RDFa. On average only three different properties (3.2) are embedded by each web sites.

At this point of the thesis, we omit the reporting of the most frequently used properties by the Microdata dataspace, as they are presented in the next chapter in detail.

Table 6.5: Total and relative amount of web sites deploying certain classes using Microdata in 2012, 2013, and 2014.

	Class	2012 #	2012 %	2013 #	2013 %	2014 #	2014 %	$\Delta_{12,14}$
1	s:WebPage	6 678	.048	69 712	.150	148 893	.182	+2.815
2	s:Blog	2 278	.016	64 709	.140	110 663	.135	+7.313
3	s:PostalAddress	19 592	.140	52 446	.113	101 086	.123	−0.117
4	s:Product	16 612	.118	56 388	.122	89 608	.109	−0.077
5	s:Article	15 718	.112	65 930	.142	88 700	.108	−0.034
6	s:Thing	587	.004	3 724	.008	80 139	.098	+22.361
7	dv:Breadcrumb	21 729	.155	44 187	.095	76 894	.094	−0.394
8	s:BlogPosting	25 235	.180	32 056	.069	65 397	.080	−0.557
9	s:Offer	8 456	.060	35 635	.077	62 849	.077	+0.272
10	s:LocalBusiness	16 383	.117	35 264	.076	62 191	.076	−0.350
11	s:Organization	7 011	.050	24 255	.052	52 733	.064	+0.287
12	s:AggregateRating	7 029	.050	36 823	.079	50 510	.062	+0.230
13	s:Person	5 237	.037	21 107	.046	47 936	.058	+0.566
14	s:ImageObject	283	.002	16 084	.035	25 573	.031	+14.463
15	s:Review	2 585	.018	13 137	.028	20 124	.025	+0.332
16	dv:Product	6 770	.048	13 844	.030	16 003	.020	−0.596
17	dv:Review-aggregate	8 517	.061	13 075	.028	14 094	.017	−0.717
18	s:Rating	1 532	.011	8 332	.018	12 187	.015	+0.361
19	dv:Offer	1 957	.014	9 298	.020	11 640	.014	+0.018
20	dv:Organization	5 853	.042	9 582	.021	10 649	.013	−0.689

Besides the strong increasing percentage of web sites making use of Microdata, which is also reflected in the usage of classes, we can find some outstanding changes. First, as already reflected by the general deployment of the vocabularies we find classes from the dv-namespace being used fewer and fewer, with the exception of dv:Offer. This class is still embedded by a stable percentage of web sites. Furthermore, classes describing web pages like s:WebPage, s:Blog and s:ImageObject are more often used. Reasons for this development are newer versions of commonly used content management systems like *Drupal* or *Typo3*. These CMS have started integrating an automatic marking up of pages created within the system.[72] Furthermore, we find relatively more web sites deploying semantic annotations describing persons and organizations in 2014, compared to 2012. The percentage of web sites providing semantic annotations about products stayed almost the same over the years.

Another remarkable observation, which can be made, is that out of the top 100 most commonly used classes, 17 are not defined by neither the schema.org nor the data-vocabulary vocabulary. Inspecting those classes shows, that they contain typos or misleading capitalization. Such data quality issues and their influence on the profiling are discussed in the following chapter.

[72]For example does *Drupal 7* make use of schema.org: https://www.drupal.org/project/schemaorg.

6.3 Related Work

Originally, the research field of data profiling is rooted in the database community. The survey by [Mannino et al., 1988] presents a comprehensive overview of the different profiling tasks, focusing on the more traditional data profiling of relational databases. Especially this survey discussed the necessity and the impact of data profiling for the task of database optimization.

In 2014, the author of [Naumann, 2014] mentions that the area of data profiling is somehow undefined and needs to be refined. He especially founds this statement on the increasing availability of data from multiple sources as well as the rise of non-relational data formats, which is in particular true for the dataspace of semantic annotations. Hence, the author tries to classify the different profiling task in to a classification schema, which separates between profiling of data from a single source and multiple sources. We have discussed the classification schema already in Section 6.1. Although the classification is quite comprehensive, we already mentioned that different relevant aspect occurring in non-relational data are currently not covered. An example is the compliance of a dataspace towards a given schema.

Besides the classification of the different profiling tasks, [Naumann, 2014] states that data profiling is in most cases driven by a use case. This use cases can be rather concrete like the usage of certain data for training a classifier or more generic like *data integration*. Especially in case of multiple data sources, data integration is the main (intermediate) challenges which needs to be considered by subsequent applications making use of a integrated dataset. Therefore, it is not surprising that the three profiling aspects, presented for "multi source" profiling by [Naumann, 2014] are closely related to the different heterogeneity, which needs to be faced in the field of data integration, namely *syntactic*, *structural*, and *semantic* heterogeneity, described in more detail in [Ozsu, 2007].

A framework, which provides various algorithms for different profiling tasks is *Metanom*, described in [Papenbrock et al., 2015]. Although the framework can handle also non-relational data, such as RDF, the covered functionalities are mainly reduced to single source profiling. Therefore, we did not make use of the tool to perform the investigations of this thesis.

6.3.1 Web Data Profiling

For the Web as a whole, there have been a number of attempts to create general profiles. One example is the `w3techs.com` web site[73], providing daily breakdowns of the top 10 000 web sites by different criteria, such as markup languages, character encoding, or content language. Unfortunately, due to the small sample of 10 000 web sites, the results of this work are questionable in terms of representativity for the whole Web. We overcome this drawback, by using a much larger sample which we think is representative for the performed studies.

[73] `http://w3techs.com/`

Other work concentrate on specific aspects, such as the structure of the underlying hyperlink graph of the Web [Meusel et al., 2015c] or the distribution of topics discussed within the content of different web sites [Chakrabarti et al., 2002a].

An investigation in the dataspace, stretched by linked open data, was recently presented in [Schmachtenberg et al., 2014]. Based on the analysis of the LOD cloud of the authors, 1 014 datasets are contained, describing various topics such as *government*, *publications*, and *life sciences*. In comparison to the overall number of web sites deploying semantic annotations, the spread of LOD is rather small. In addition, the variety of covered topics is rather limited. We discuss further related work analyzing the quality of data retrieved from the LOD cloud in Chapter 7.

6.3.2 Microformats, RDFa and Microdata

The data provided by the realization of the vision of the *semantic web* by semantic annotations in HTML pages has first been studied by [Mika, 2011] in 2011. The authors analyzed the adoption of the three semantic markup languages Microformats, RDFa, and Microdata based on a corpus obtained by the Yahoo! crawler. They refined their results in a later study, using a corpus obtained by the Bing crawler [Mika and Potter, 2012]. Their study focused solely on the number of web pages and web sites using the different formats. In addition, the authors analyzed the number of statements (quads) which could be extracted from the web pages. This study was omitted by us so far, as the results are bias based on the selection strategy which was employed to collect the corpus. The data which was used in [Mika, 2011] is not publicly available. In the same year, [Mühleisen and Bizer, 2012] presented their first study on the deployment of the three mentionedsemantic markup languages based on a part of the 2012 corpus of CC. Besides general statistics about the overall adoption, a more fine-grained analysis about class distributions and deployment of classes is depicted, which we have refined in this chapter.

This study formed the foundation for the *Web Data Commons* (WDC) project, which until now, and as part of this thesis, has continued profiling the deployment of semantic annotations based on the public web corpora of CC [Bizer et al., 2013, Meusel et al., 2014b]. The outcomes of those profiling activities are manifold. First, especially in one of our previous studies, it has been shown that in contrast to entities contained in LOD, which are linked to each other, in the dataspace of semantic annotations links between entities barely exist. Furthermore, the results of former studies already indicated the increasing adoption of Microdata and the importance of thereby deployed semantic annotations, which we again discovered in the findings of this chapter. Especially in combination with schema.org, more and more web sites deploy semantic annotations. This vocabulary is promoted by the major search engine companies Bing, Google, Yahoo!, and Yandexand is continuously updated in order to improve the topical coverage [Guha, 2014]. Besides the lack of links between entities, it has also been shown that HTML pages annotate the entities on the page with only a few properties. This problem of flat Microdata annotations has been analyzed more deeply for the class `s:Product` in [Petrovski et al., 2014].

With *flat*, the authors denote that only a small number of properties is used to describe an entity, although the schema potentially offers a larger set of properties to annotate information on a fine-grained level. Further, the authors mention that within many HTML pages more detailed information are mostly summarized within the property `s:description`. Therefore, in order to enable the full potential of product-related semantic annotations the fine-grained information needs to be identified. The authors propose to use regular expressions for extracting features from the title and the description of products marked up with Microdata.

All the works mentioned so far present mostly empirical studies of the current deployment of the different semantic markup languages and schema without a discussion of possible discrepancies and potential flaws of such an analysis.

Moving the focus of the analysis towards the underlying data and its influence on the outcome of the profiling task, [Mika and Potter, 2012] mentioned that the results of their studies are based on different datasets and that they might not be directly comparable. In our analyses we make use of different corpora, which were collected in different years. For all three corpora, a similar selection strategy, based on the popularity of web sites was used. We therefore think, that the results from the different corpora are comparable. Another work, presented in [Stolz and Hepp, 2015] analyzes how representative, with respect to the deployment of semantic annotations, the corpora provided by the Common Crawl Foundation are. They compare the results from former WDC studies with results obtained by crawling the sitemaps. By considering a set of 100 different web sites, they reveal that the underlying crawling strategy of the used corpora can have an influence on the outcome of the profiling. As the representativity is an important aspect when using samples, we have comprehensively discussed the issue in Chapter 5, and concluded that the data, which is used by WDC as well as in this thesis, is representative for the (public) Web.

6.4 Summary

Within this chapter, we generated a first profile of the dataspace of semantic annotations in HTML pages, embedded using Microformats, RDFa, and Microdata. We have compared the overall and markup language-specific adoption based on three corpora from 2012, 2013, and 2014. We have identified an overall increasing adoption of semantic annotations in the Web, where especially the semantic markup language Microdata becomes more and more popular. The deployment of semantic annotations by Microformats slightly decrease, but are still dominant in the Web, especially the markup language MF`hCard`.

Although the vocabularies mostly offer a rich variation of properties to describe certain classes, only a few properties are used by the web sites to mark up their information on average. The diverse topical coverage of semantic annotations, deployed by the different semantic markup languages, makes the dataspace interesting for various applications and use cases.

In order to gain more fine-grained insights into the dataspace it is necessary to use markup language-specific methods in order to estimate the quality of the data, as well as its data overlap. Also we have seen an increasing adoption of Microdata, an important aspect is the evolution of the vocabulary and the actual adoption. Insights into those aspects allow a dedicated analysis of the utility of the dataspace for particular use cases. The both aspects are discussed in the following chapters in more detail.

Chapter 7

Profiling of schema.org Microdata

Within the previous chapter, we have analyzed the overall adoption of the dataspace of semantic annotations in the Web. The profile was generated with respect to the percentage of web sites embedding semantic annotations by a particular semantic markup language, vocabulary or class, using the three corpora extracted from web corpora.

As already mentioned in the previous chapter, the underlying profiling methodology did not make use of any further knowledge of the dataspace, e.g., vocabulary definitions or RDF-based restrictions, and the results are thereby partly superficial.

Within this chapter, we want to gain further, more fine-grained insights in the dataspace of semantic annotations. We focus especially on aspects of *compliance* and *duplicate content*, and analyze both with respect to their influence on the profile of the dataspace.

Compliance is one of the major issues within the Web. Basically it describes all issues where the actual usage of a standard violates the schema (definition of the vocabulary). One reason for the occurrence of compliance issues in the Web is the large number of contributors and their diversified knowledge of the standard, already mentioned by [Hernández and Stolfo, 1998]. In the context of semantic annotations, we focus on schema compliance. This includes all issues where the usage of a schema or vocabulary does not follow its definition. Violations can arise from simple typos within the name of the class or properties and going further to the usage of undefined properties with certain classes. The detection and potential correction of such issues support applications to correctly *understand* the data and make use of it.

Duplicate Content is another issue of the Web in general and also influences the dataspace of semantic annotations embedded by Microformats, RDFa, and Microdata. One of the simplest examples of duplicated content on two or more

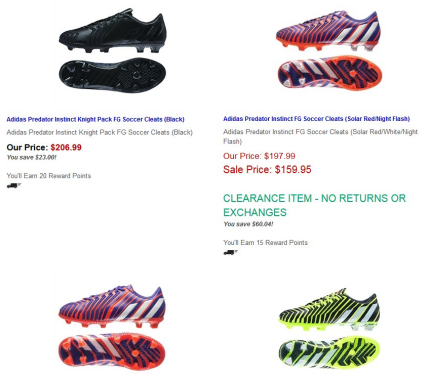
web pages are information contained in the *header* and *footer* of pages from the same web site. Also web site-comprehensive examples exist, were two different pages just display one and the same content, e.g., an advertisement provided by a third-party server or database. Therefore, the assumptions is obvious that whenever semantic annotations are marked up in such parts of the web page, the data corpus contains duplicates. Again, the detection and removal of duplicated entries allows the creation of a more accurate profile of the dataspace. Especially with respect to the number of different entities, e.g., the number of different product information provided by a web site, this issue should be taken into account for profiling.

In order to overcome obstacles arising from the two mentioned issues, we incorporate rather superficial pre-processing methods before the actual profiling. The used methods make use of the schema as well as a straightforward assumption about duplicates within RDF-graphs. Therefore the methods can easily be applied to other vocabularies and parts of the dataspace.

Summarizing, the contribution of this chapter is three-fold. First, while the previous investigations show aggregated numbers, this chapter attempts to estimate the total number of entities in the schema.org Microdata dataspace by class, thus adding another important facet to the topical profile of semantic annotations. Second, while the previous statistics about Microdata rather focus on *adoption*, i.e., on data providers, this chapter changes the perspective to the data usage. We exploit duplicate detection and focus on the number of *entities* which are described by the schema.org Microdata dataspace. Third, we more closely examine the interaction of data cleaning and data profiling for web data. It is important to note that the contributions do not focus on the analysis of the number of entities which are described by the dataspace, as such results are misleading due to the dynamic of the Web. The contributions rather focus on the demonstration of the influence of dataspace-specific data cleansing methods towards the overall profile. Furthermore, they showcase, how semantic annotations need to be processed before they can be used by an application.

Therefore, within this chapter, we explain the facets of the two mentioned aspects in more detail in Section 7.1 and present in the following section a methodology to overcome the identified obstacles. Section 7.3 presents the results of the methodology and shows the influence on the profile created for the dataspace. Within the last two sections the results are first discussed due to limitations of the methodology and the overall outcome is then summarizes.

Parts of the methodology, results and parts of the discussion have already been published in [Meusel and Paulheim, 2015] and [Meusel et al., 2016].



(a) Example web page with annotations.

```
_:n1 rdf:type s:Product .
_:n1 s:name "Predator Instinct
                    Knight Pack FG" .
_:n1 s:brand "Adidas" .
_:n1 s:price "206.99" .

_:n2 rdf:type s:Product .
_:n2 s:name "Predator Instinct FG" .
_:n2 s:brand "Adidas" .
_:n2 s:price "159.95" .

_:n3 rdf:type s:Product .
_:n3 s:name "Predator Instinct FG" .
_:n3 s:brand "Adidas" .
_:n3 s:price "206.99" .

_:n4 rdf:type s:Product .
_:n4 s:name "Predator Instinct FG" .
_:n4 s:brand "Adidas" .
_:n4 s:price "189.95" .
...
```

(b) Subset of extracted triples.

Figure 7.1: A fashion vendor web page, and an example set of triples extracted from that web page.

7.1 Problem Statement

As we have shown in [Meusel et al., 2014b, Meusel and Paulheim, 2015], semantic annotations embedded with schema.org Microdata reveal several challenges which need to be taken into account. Those challenges can be categorized into two general aspects of data cleaning: *identity resolution* or *duplicate detection*, and *schema compliance*.

To understand those issues we consider the example depicted in Figure 7.1. In the example a number of products have been annotated with a prize and a name.

When creating a profile of all the data provided at the vendor's web sites, e.g., to answer the question how many products by how many brands are sold by the vendor, some challenges arise. First, information may be duplicated: the same product may appear on different web pages (i.e., overview pages, detail pages, special offer pages). While such a content duplication may be deliberate and useful, it is an issue that needs to be addressed for data profiling. Since the data model only uses blank nodes, there are no unique identifiers.[74] Thus, it is required to first detect duplicates before answering the initial question of how many products are sold by the vendor.

Moreover, the set of extracted statements shown in Figure 7.1b is not compliant to the data model of schema.org. For example, the property `s:brand` expects an object of class `s:Brand`, not a string value. Likewise, `s:price` is not supposed to be attached directly to an object of class `s:Product`, but to an offer. Thus, any state-

[74] Note that simply replacing each blank node identifier with a unique URI would not remedy that problem, since the same real-world object would then be referred to by a multitude of URIs.

ment about the number of entities of the class `s:Brand` and `s:Offer` would be influenced by this non-compliance to the data model. In [Meusel and Paulheim, 2015], we have shown that a non-negligible amount of web sites is affected by such issues.

In the following, we will provide deeper insights into both issues.

7.1.1 Duplicates

Duplicates can lead to skewed statistical profiles of the data. There are different causes for duplicates, introducing different biases. As we are dealing with data extracted from crawled web pages, various phenomena can lead to duplicates:

Web Page Headers and Footers For business web pages, the page title, header or footer often contain the company name plus further information like the address. Since this content is replicated across all pages, it leads to new entities for each page of the web site. An example can be found in Figure 7.2a, where a B&B website annotates the header information, which is included on every page of the site.

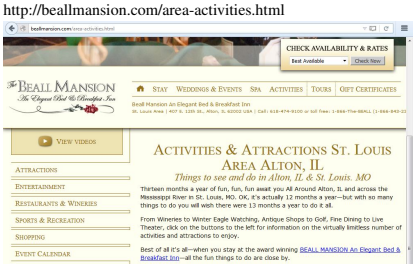
Content Duplication Many web pages are created from databases and/or content management systems. Hence, the same piece of content can appear on various pages. In the example depicted in Figure 7.1, each offer for a particular shoe can occur in various pages (product lists for multiple categories, detail pages, special offer pages). An example is shown in Figure 7.2b. *Meetup*, an online community finder annotates the information (e.g., communities) on the overview page as well as on the detail page of the community. In the case of mashups, content duplication may also occur across web sites. This is, e.g., often observed for blog contents which are replicated across different sites.

Replication of Web Sites Some web sites operate under multiple domains with identical contents. Among others, this is the case for shops that also operate under common misspellings of their domain, or domains operating under different top level domains. Figure 7.2c shows the German as well as the Austrian web page version displaying the same product. Thus, all the data extracted from those web sites is also duplicated.

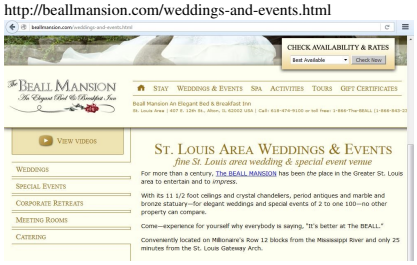
Non-canonical or Similar URLs Crawlers may also follow URLs that contain arguments, e.g., for searching products or displaying products from certain categories. These may lead to similar or nearly similar pages under the same URL. Likewise, the same page may exist under different URLs on a web site, e.g., for search engine optimization or marketing channel tracking. An example is shown in Figure 7.2d, displaying the same web page of the *2modern* online shop solely with different fragments within the URL query.

Within the four mentioned examples, the information extracted from each of the two shown web pages (partly) overlap and lead to duplicated entities within the dataspace.

http://beallmansion.com/area-activities.html

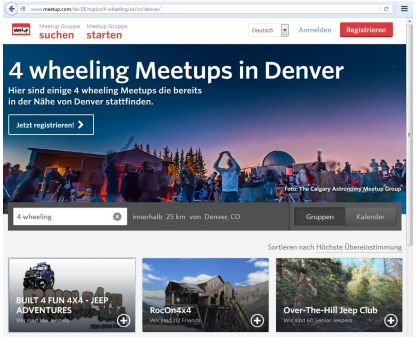


http://beallmansion.com/weddings-and-events.html

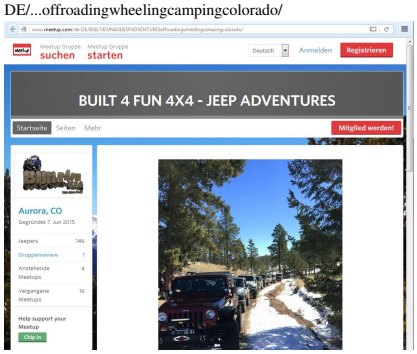


(a) B&B web pages with identical header.

http://www.meetup.com/de-DE/topics/4-wheeling/us/co/denver/

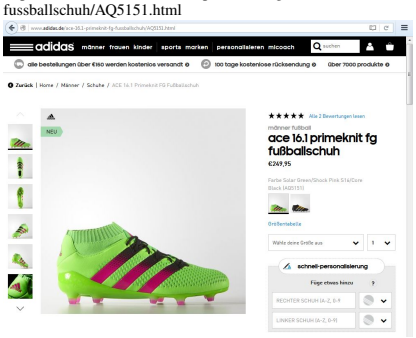


http://www.meetup.com/de-DE/...offroadingwheelingcampingcolorado/

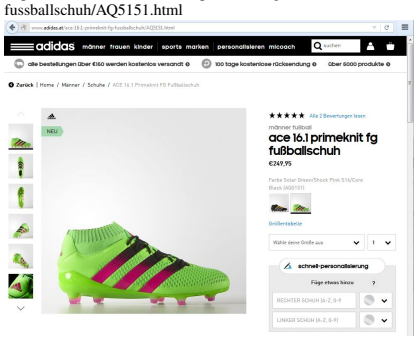


(b) Overview and detail meetup web page.

http://www.adidas.de/ace-16.1-primeknit-fg-fussballschuh/AQ5151.html



http://www.adidas.at/ace-16.1-primeknit-fg-fussballschuh/AQ5151.html



(c) Adidas soccer boot web pages from German and Austrian adidas domain.

http://www.2modern.com/collections/bobles

http://www.2modern.com/collections/bobles?ck=x1f9Lai-ASBY711B

(d) Identical shopping result web pages using different URL query fragments.

Figure 7.2: Example web pages illustrating different reasons for duplicated entities extracted from HTML pages.

It is noteworthy that in all those cases, content (and hence, semantic annotations) duplication is deliberate and desirable from the web site providers' perspective. However, that content duplication leads to a data duplication in the extracted corpus of RDF data, which is an obstacle to accurate data profiling.

Besides all the issues leading to duplicated entities, there are circumstances where entities seems to be duplicates (described by exactly the same property-value pairs) but indeed do *not* refer to the same real-world object.

Those come into existence since the data markup is in many cases not complete. In the example depicted in Figure 7.1, the duplicates may refer to the exact same shoe or to the same shoe but in different sizes and colors. Since neither size nor color is marked up, it is impossible to tell one apart from the other.

Nevertheless, the examples show that simply counting occurrences of class instantiations cannot lead to reliable estimates of class distributions. Instead, a duplicate detection, or identity resolution, is required to produce reliable estimates. Since our manual estimation showed that the vast majority of duplicate RDF nodes are actually duplicate representations of the same real-world object, we pursue this approach to arrive at a more accurate (topical) profile of the schema.org Microdata dataspace.

7.1.2 Non-compliance to the Schema

The example depicted in Figure 7.1 contains some violations of the schema.org data model. Such violations are quite frequent, and they can significantly skew estimations of class distributions. In [Meusel and Paulheim, 2015], we have analyzed the most common deviations based on the number of data providers of actually deployed Microdata from the defined schema, all of which lead to a skewed view on the data:

Usage of Undefined Classes and Properties To a large extent, undefined classes and properties are the result of misspellings of actually existing classes and properties. Correcting the spelling automatically, where possible and appropriate, is required to come up with reliable data profile estimates. A large fraction of misspellings is due to wrongly capitalized class names (e.g., `http://schema.org/postaladdress` instead of correctly capitalized `http://schema.org/PostalAddress`) or the names of properties (e.g., `http://schema.org/URL` instead of `http://schema.org/url`). Figure 7.3a shows a *Yellow pages* page annotating their entries using the lower-cased class `http://schema.org/localbusiness` instead of the correct class `http://schema.org/LocalBusiness`.
It is important to state that schema.org provides an extension mechanism to introduce new classes and properties, where the new classes are added as a suffix of the superclass e.g., `http://schema.org/VideoGame/MMORPG`. Such occurrences are *not* counted as undefined classes.

Object Properties with Literal Values Object properties are properties that require an entity in the object position, not a literal value. In Figure 7.1, the property `s:brand` is such an example, which actually requires an entity of type `s:Organization` or `s:Brand` in the object position. Although the statement can be understood by a human (and, to a certain extent, also by a fault-tolerant agent), this example would lead to an under-estimation of the number of organizations and brands in the dataspace. Figure 7.3b depicts a product page of an apparel online shop which directly annotates for the property `s:aggregateRating` the value of the product rating as literal, instead of introducing the entity of class `s:AggregateRating` with its property `s:ratingValue`.

Property Domain Violations While most properties come with a domain definitions, i.e., valid classes that can be used in the subject position, they are not always respected by data providers. In many cases, those are shortcuts, which omit an intermediate concept, as shown in Figure 7.4. On the left side, the original information extracted directly from the markup in the HTML page is shown. Here the property `s:ratingValue` is directly used for `s:Product`, although its domain is `s:AggregatedRating`, i.e., an entity of type `s:AggregatedRating` would be needed (see right side), but has been omitted by the data provider. Those shortcuts lead to an underestimation of the class of the omitted intermediate instance. A real world example is shown in Figure 7.3c, where the property `s:ratingValue` is directly used for the class `s:Product`.

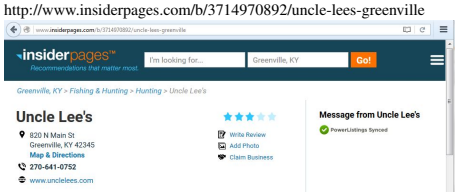
In all those examples, given that a certain violation is reasonably frequent, it can again skew the distribution of classes significantly. In Section 7.2, we show that most of the violations can be fixed by reasonable heuristics, e.g., by automatically creating missing entities. Applying those heuristics does not only lead to schema compliant data, but, at the same time, also changes the distributions and profiles created for the data.

There are various possible reasons for non-compliance with the schema. Apart from simple mistakes (such as typos) made by web developers, we have identified some additional common causes [Meusel et al., 2015a]:

Disrespect for Deprecations Since schema.org is constantly revised and refactored, some classes and properties occasionally become deprecated. As we show in Chapter 8, it sometimes takes time for those deprecations to be adopted by data providers (and also data consumers, which may also lead to an intentional use of deprecated constructs).

Pre-standardization Usage In a few cases, new properties and classes are used before they become officially included in the standard, e.g., since they were already discussed on mailing lists as *potential* new classes or properties.

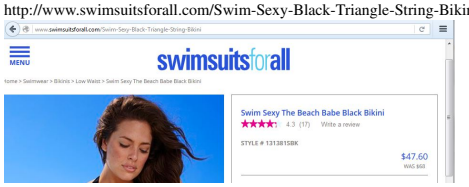
http://www.insiderpages.com/b/3714970892/uncle-lees-greenville




```
_:n1 rdf:type s:localbusiness .
_:n1 s:name "Uncle Lee's" .
_:n1 s:address _:n2 .
_:n2 rdf:type s:PostalAddress .
_:n2 s:addressLocality
              "Greenville" .
_:n2 s:postalCode "42645" .
...
```

(a) Yellow pages page using wrong capitalized schema.org class.

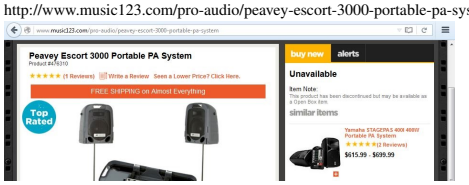
http://www.swimsuitsforall.com/Swim-Sexy-Black-Triangle-String-Bikini




```
_:n1 rdf:type s:Product .
_:n1 s:name "The Beach Babe
                  Black Bikini" .
_:n1 s:brand "Swim Sexy" .
_:n1 s:aggregateRating "4.29" .
_:n1 s:price "47.60" .
```

(b) Apparel shopping page using object properties with literals.

http://www.music123.com/pro-audio/peavey-escort-3000-portable-pa-system




```
_:n1 rdf:type s:Product .
_:n1 s:name "Escort
      3000 Portable PA System" .
_:n1 s:brand "Peavey" .
_:n1 s:sku "476310" .
_:n1 s:ratingValue "5.0" .
_:n1 s:reviewCount "1" .
...
```

(c) Music instrument shopping page using shortcuts.

Figure 7.3: Example web pages illustrating most frequent deployed schema violations.

Recontextualization Properties are quite frequently used in ways for which they were originally not intended, e.g., with classes that they are not defined for. If such recontextualizations occur frequently, they often hint at shortcomings of the standard, i.e., information that data providers would like to express, but cannot according to the schema. Again, in many cases, the schema can be adapted to allow the recontextualization.

Furthermore, most online validators check for correct Microdata syntax, but are partially agnostic to the schema as such in their validation. An example is the *Structured Data Testing Tool* by Google[75]. As of January 2016, besides a Microdata syntax validation, the tool offers a partial validation of the schema, which reports domain violations, but no range violations, e.g., literals used for object properties. In addition, the tool is tailored to support the correct annotation for *Google's Rich Snippets* and enforces the use of the `s:price` within an `s:Offer`, although it is not mandatory in the schema. Such tools can help web developers to create (almost) compliant annotations, but in order to annotate all available data correctly deeper knowledge of schema.org is required.

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

```
_:n1 rdf:type s:Product .
_:n1 s:ratingValue "5" .
```

(a) Original RDF statements extracted from HTML page.

```
_:n1 rdf:type s:Product .
_:n1 s:aggregatedRating _:n2 .
_:n2 rdf:type s:AggregatedRating .
_:n2 s:ratingValue "5" .
```

(b) Corrected RDF statements extracted from HTML page.

Figure 7.4: Example of a *shortcut* using directly a property for a class, which is not included in the properties domain definition.

7.2 Methodology

As discussed above, semantic annotations embedded in HTML pages using Microdata have some quality issues, in particular due to duplicates and schema violations. Thus, in order to generate a more accurate profile of the dataspace in terms of the number of web sites embedding semantic annotations for certain classes and the number of different real-world objects described by the web sites, we need to first address those quality issues.

In this section, we therefore introduce a multi-step pipeline which subsequently addresses those issues through duplicate removal and heuristically fixing schema compliance violations. Based on that pipeline, we create a profile of the data corpus after each step to analyze the impact of each step. In addition, after having applied simple but highly effective data cleaning steps, we further try to more accurately estimate the number of unique real-world objects described by the entities included in the corpus.

Possible flaws of the subsequent presented methods are discussed in detail in Section 7.4 in order to not interrupt the flow of reading.

7.2.1 Syntactic Duplicate Removal

In our approach, we make use of a syntactic duplicate removal strategy. This strategy is applied per web sites and, later in the pipeline, also globally.

In order to do so, we follow the W3C's RDF semantics[76] definition of RDF document equivalence, which states that two RDF graphs are equivalent if there exists an isomorphism between the two graphs that maps each resource and literal to itself, and each blank node to another blank node.

As stated before, all RDF extracted from Microdata forms a directed acyclic graph. Thus, each entity can be described as a graph constituted by the tree it spans. In our syntactic duplicate removal step, we compare the graphs spanned by all entities in the RDF graph extracted from each page. First, this step is applied by comparing all entities from one web site. Later, we also apply it across web sites.

[76] http://www.w3.org/TR/rdf11-concepts/#graph-isomorphism

To implement the duplicate removal in a scalable fashion, we represent each graph as a string. Since the graphs are free of cycles, we can enumerate all paths from the root to the leaves in an ordered way, where we alphabetically order by property name and property value. Those paths can be represented as strings, with the same token being used for each blank node. By concatenating those tokens into a string, again following an alphabetical ordering, we can compare two entities efficiently by comparing a hash function over those strings.

7.2.2 Heuristics to Correct Schema Violations

Based on the findings of our previous work about the most commonly made errors when deploying schema.org Microdata, we apply a set of heuristics that we proposed in [Meusel and Paulheim, 2015] to address violations of the schema and render schema compliant data.

Identifying and Fixing Wrong Namespaces

According to our observations, most namespace violations are the result of wrong capitalization, e.g., using `http://SCHema.org` instead of `http://schema.org`, leading `www.` and the use of the `https` protocol instead of the non-secure version, or using too few or too many slashes, e.g., `http://schema.orgStore` instead of `http://schema.org/Store`. Therefore, we apply the following heuristic to correct those violations which we already proposed in [Meusel and Paulheim, 2015]:

1. Removal of the leading `www.` before `schema.org`
2. Replacement of `https://` by `http://`
3. Conversion of the whole domain name to lower case
4. Removal of any additional sequence between `http://` and `schema.org`
5. Addition of an extra slash after `schema.org`, if none is present

Handling Undefined Classes and Properties

In most cases, undefined classes and properties are not freely made up by the data providers, but are the result of misspellings. According to our findings, the vast majority of those issues are a result of wrong capitalization (e.g., `s:contentURL` instead of `s:contentUrl`). Thus, whenever parsing Microdata entities from web pages, we suggest to not take capitalization into account, and replace each schema element with the properly capitalized version.[77]

[77] Note that schema.org does not define any pair of classes or properties that differ only in capitalization. There exist, however, pairs of a property and a class which differ only in capitalization, but since within Microdata, classes and properties are annotated with different properties within HTML (namely `itemtype` and `itemprop`), they cannot be mixed up.

Handling Object Properties with a Literal Value

The main three objects which are modeled by web masters as string literals are `s:Organization`, `s:Person`, and `s:PostalAddress`, although the schema expects the use of an object in those positions. Based on a manual inspection of over 700 randomly chosen literal values for the three properties `s:author`, `s:creator`, and `s:address`, we see that the majority of literals for `s:Person` and `s:Organization` are person/organization names or URLs, while literals for `s:PostalAddress` are usually represented a textual representation of the address.

From this observation, we derive the following strategy for fixing literal valued object properties: Given a triple

1. `_:1 s:op l .`,

where `s:op` is an object property, and `l` is a literal, replace the triple by

1. `_:1 s:op _:2 .`
2. `_:2 a s:t .`
3. `_:2 (s:name|s:url) l .`

Here, `s:t` (representing the class of the object) is the range of `s:op`, or the least abstract common superclass of all ranges, if there are more than one. If `l` is a valid URL[78], it is set as the `s:url` of the newly created entity. Otherwise, it is used as its `s:name`.[79]

Handling Property Domain Violations

Another common error is the usage of properties on classes that they are not defined on. As discussed above, a common cause for this are *shortcuts* taken by the data provider as shown in Figure 7.4.

In order to expand the wrong triple to the correct set of triples, we need to guess what the data provider meant. To that end, we use the following approach: Given two triples

1. `foo:x s:r foo:y .`
2. `foo:x a s:t`

where `s:t` is *not* the domain of `s:r`, we try to find a relation *R* and a type *T* within schema.org such that one of the following two patterns is fulfilled:

1. *R* `s:domainIncludes s:t .`
2. *R* `s:rangeIncludes` *T* `.`
3. `s:r s:domainIncludes` *T* `.`

1. *R* `s:rangeIncludes s:t .`
2. *R* `s:domainIncludes` *T* `.`
3. `s:r s:domainIncludes` *T* `.`

[78] Following `http://www.ietf.org/rfc/rfc1738.txt`.

[79] Note that `s:name` is more generic than, e.g., the name of a person. It is comparable to `rdfs:label` in RDF.

If there is *one unique* solution for *only one of the two* pattern, we replace the erroneous triple with the solution we detected. Based on the observations in [Meusel and Paulheim, 2015], we expect to find a unique solution for around 30% of the shortcuts where over 45% will have multiple solutions and therefore will not be corrected. In a subsequent step, we unify all newly created entities of one class into one entity.

Thus, extending the example in Figure 7.4 with the property `s:ratingCount`, results in the following triples:

```
1. _:n1 rdf:type s:Product .
2. _:n1 s:ratingValue "5" .
3. _:n1 s:ratingCount "10" .
```

Applying the schema correction step of our proposed methodology would lead to:

```
1. _:n1 rdf:type s:Product .
2. _:n1 s:aggregateRating _:n2 .
3. _:n2 rdf:type s:AggregateRating .
4. _:n2 s:ratingValue "5" .
5. _:n1 s:aggregateRating _:n3 .
6. _:n3 rdf:type s:AggregateRating .
7. _:n3 s:ratingCount "10" .
```

When applying an additional duplicate detection step and unifying the created entities, the second occurrence of `s:AggregateRating` will be merged into the first one:

```
1. _:n1 rdf:type s:Product .
2. _:n1 s:aggregateRating _:n2 .
3. _:n2 rdf:type s:AggregateRating .
4. _:n2 s:ratingValue "5" .
5. _:n2 s:ratingCount "10" .
```

7.2.3 Combined Approach

In order to combine the different cleansing steps described above and measure their impact on the profile of the resulting data corpus, we propose a five step pipeline as depicted in Figure 7.5:

1. Starting from the raw WDC Microdata corpus, we filter all the entities which define statements in the namespace *schema.org*. The resulting dataset is referred to as $S1$.

[80] Please note that the size of the *datasets* in this figure does not represent the relative sizes of the real datasets and thus cannot be used for any estimations in reduction.

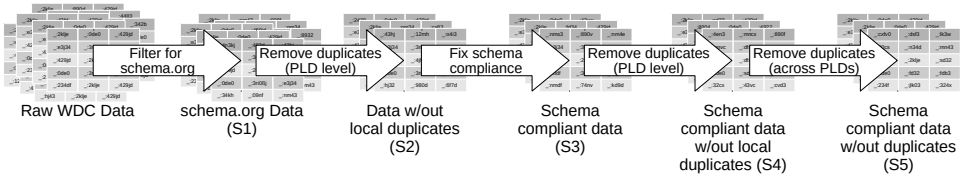


Figure 7.5: Combined cleansing pipeline overview with the produced intermediate data corpora.[80]

2. Using the statements in $S1$, we run the syntactic duplicate removal for all statements within one web site. This step removes duplicated entities which are crawled due to duplicated pages/URLs, repeated content within a web site, footers, etc. The resulting dataset is referred to as $S2$.
3. The next step is to apply the heuristics for enforcing schema compliance. Here, property and class names are changed to their correct spelling, entities are introduced where omitted, and *shortcuts* are expanded where possible. The resulting dataset is referred to as $S3$.
4. As shown above, applying the heuristics leads to new entities, therefore, it is also possible that new duplicates are introduced. Thus, a second duplicate removal per web sites is performed. The resulting dataset is referred to as $S4$.[81]
5. In a final step, duplicates are also removed across web sites. The resulting dataset is referred to as $S5$.

The latter two steps are separated because $S4$ allows a profiling using the number of web sites as a unit of measure (e.g., how many web sites provide entities of a certain class), while $S5$ only contains statements about entities. Thus, in order to allow statistics on both levels of aggregation, the steps are separated.

In the following, we apply the pipeline and create a topical profile (i.e., analyze the most frequently deployed classes and properties) after each step. Thereby, we analyze how the different steps of the pipeline influence the created data profile, and subsequently move towards a more realistic profile.[82]

7.2.4 Semantic Identity Resolution

After the execution of the pipeline, we have removed syntactical duplicates from our corpus as well as fixed a set of schema violations. Within a last step, we are interested in finding entities which describe the same real-world object, although they are syntactically different. This is a problem which is very different from

[81] The result would be equivalent if the first duplicate removal step from S1 to S2 was omitted. However, since that first step removes a lot of duplicates, the overall processing is faster with the proposed pipeline.

[82] We provide the source code which was used to remove duplicates as well as to apply the heuristics in order to fix schema violations as part of the WDC project: `http://webdatacommons.org/structureddata/2014-12/cleansing.html`.

Table 7.1: Selected classes with (pseudo-)key properties.

Class	(Pseudo)-Key Property (pk)
s:Article	s:articleBody
s:Blog	s:url
s:BlogPosting	s:articleBody
s:ImageObject	s:contentUrl
s:LocalBusiness	s:taxID, s:vatID
s:Offer	s:mpn, s:gtin{8,12,13,14}
s:Organization	s:taxID, s:vatID
s:Person	s:taxID, s:vatID
s:Product	s:mpn, s:gtin{8,12,13,14}
s:WebPage	s:url

purely syntactical duplicate removal and requires class-specific approaches, thus, we do not directly include it in the presented pipeline.

The general assumption to find out which entities refer to the same real-world object is, that depending on the class some properties and their values exist, that can be considered as globally valid identifiers. We hence call those properties (pseudo-)key properties (pk).[83] For some classes, those properties are obvious to find, for example the *taxID* for persons and organizations. But for other entities some deeper domain knowledge is required. For example entities of the class `s:Product` can be annotated with the *Global Trade Item Number* (gtin), as well as the *Manufacturer Part Number* (mpn). As shown in Table 7.1, we manually identified potential pks for the 10 most frequent classes in schema.org (representing over 70% of all included entities).

Unfortunately, as later analysis will show, not all entities of a class make use of the identified pks. In fact, only a small subset of entities belonging to these classes provides values for those properties. We therefore cannot directly calculate the number of referenced real-world objects but have to estimate it based on the ratio of entities in the subset describing the same real-word object making use of the identified properties. To do so, for each of the selected classes E, we extract the subset of entities E_{pk} (of this particular class) which are annotated with the respective pk. We further identify the subset of duplicated entities $E_{pk,dup} \in E_{pk}$, meaning those entities which have a non-unique value for pk. As a consequence, we can estimate the number of duplicates beyond the overall number of entities of a certain class, based on the probability of an entity being duplicate within E_{pk}, which is $P(dup) = \frac{|E_{pk,dup}|}{|E_{pk}|}$.

Based on the number of estimated duplicates, we can further calculate the number of *unique* entities among the duplicates by projecting the ratio between the number of duplicates $|E_{pk,dup}|$ and the number of unique pk-values within $E_{pk,dup}$. From these results, we can calculate the estimated reduction of *unique* entities.

[83] In the Web Ontology Language OWL, those would be coined *inverse functional properties* – however, such properties are not defined in the schema.org standard.

Example: In the following, we use the class `s:Blog` as example in order to demonstrate how we apply the computations in practice (which are presented later in detail in Table 7.8). After applying the cleaning steps, we counted $2\,384.2k$ entities for this class (E). Of those, $388k$ make use of the identified pk property `s:url` (E_{pk}). Among them, we find $79.k$ entities sharing $11.7k$ different non-unique pk property values ($E_{pk,dup}$). Based on these numbers, we calculate the probability for an entity to be a duplicate $P(dup) = \frac{79.7k}{388k} = 0.21$ and in addition, the average number of entities with the same pk property value $\frac{79.7k}{11.7k} = 6.79$. Thus, we can estimate the number of duplicates in E, which is $E_{dup} = P(dup) * |E| = 0.21 * 2\,384.2k \cong 489.6k$. Incorporating the average number of entities with the same pk property value, we can estimate that $\frac{489.6k}{6.79} \cong 72.1k$ unique entities are contained in the $489.6k$ duplicates. In conclusion, we estimate the number of unique entities within E as the number of non-duplicated entities plus the estimated number of unique entities in E_{dup} which is $1\,894.5k + 72.1k \cong 1\,966.7k$.

It is important to mention, that we can only estimate and not count the unique number for *all* entities of a class, since not all entities make use of the pk property.

7.3 Empirical Findings

The total number of entities, classes, and properties, as well as the number of RDF statements of the five different generated datasets (i.e., $S1 - S5$) is described in Table 7.2. During the first four steps ($S1 - S4$), the duplicate removal within the web sites reduces the number of entities by almost 60%. Using the heuristics to fix domain violations, a decent number of entities is added to the corpus. These entities have been included implicitly before. Detecting duplicates within the whole corpus, the number of entities is again reduced slightly. It is furthermore remarkable that the schema compliance heuristic removes more than 800 undefined classes and almost $5\,000$ undefined properties.

Table 7.2: Number of quads, entities, unique classes and properties contained within each of the five created datasets.

	S1	S2	S3	S4	S5
# RDF statements (in millions)	6 778	3 302	3 805	3 389	3 019
# Entities (in millions)	1 543	609	862	655	599
# unique Classes	2 664	2 664	1 801	1 801	1 801
# unique Properties	24 340	24 340	19 552	19 552	19 552

7.3.1 Syntactical Duplicate Removal and Correction of Schema Violations

Table 7.3 contains the 40 most commonly used classes by the number of web sites making use of the class in $S1$. The numbers are calculated using a materialized class hierarchy, i.e., including all subclasses of the particular class.

It is important to note that the increase in the total number of web sites making use of `s:Thing` – which in the end includes all classes from all web sites– is *not* mainly due to web sites which do not make use of classes at all. It is rather due to web sites making use of only *undefined* classes, which are not counted as subclass of `s:Thing` in $S1$.

Since the numbers are aggregated by web site and duplicate removal across web sites is only done at the transition from $S4$ to $S5$, the numbers cannot decrease between $S1$ and $S4$. Furthermore, immediate numbers (e.g., of $S2$ or $S3$) are not stated as they are equal to those of $S1$ respectively $S4$ because the number of web sites is only affected by the introduction of new RDF statements or the transformation of class names into their correct spelling. For $S5$, a breakdown by web site does not make sense. Once duplicates are removed across web sites, it is no longer clear for which web site to count a class or property.

Overall, we observe an increase in the amount of web sites providing semantic annotations for a certain class that varies between 1% and 189%. Remarkable increases can be observed for `s:WebPageElement` as well as for `s:Store`. For `s:WebPageElement`, this results from applying the heuristic to fix literal values of the object property `s:mainContentOfPage`. For `s:Store`, the effect is due to namespace violations, as there are no less than $6\,390$ web sites using the "class" `http:/schema.orgStore` (with two missing slashes), which is fixed by our heuristic.[84]

Table 7.4 contains the 40 most commonly used properties, based on the number of web sites deploying them within $S1$. Again, only the numbers for $S1$ and $S4$ are shown. Similar to the classes, we observe increasing numbers, although some changes are marginal. The most outstanding increase can be observed for `s:headline` and `s:contentURL`, which are both increased by the correction of former typos.

Table 7.5 and Table 7.6 list the 40 most common entities within the five different steps of the proposed pipeline ordered by the number of entities in $S1$. The numbers are again calculated using a materialized class hierarchy. We omit to report the numbers for $S3$ as they only differ marginally from those for $S4$. For each step of the pipeline, we calculate the difference Δ to the previous step as well as the difference Δ to $S1$. Therefore, the last column of the table includes the overall changes of the number of entities caused by applying the full pipeline.

In the first row of Table 7.5, showing the number of all entities (as `s:Thing` is the superclass of all classes defined in schema.org) we can directly observe the

[84] Inspecting a sample of sites deploying the class `s:Store` with the mentioned namespace violation we found that a large fraction is created by an e-commerce software system offered by *Volusion* (`http://www.volusion.com/`), and probably caused by a bug in this software. Affected web sites are, besides others those of `4goslep.com`, `armynavyusa.com`, `bucklecity.com`, and `dollmarket.com`.

Table 7.3: Top 40 most commonly deployed classes, including their subclasses, ordered by the number of web sites within $S1$.

	Class	S1	S4	Δ %
1	s:Thing	703 623	725 474	0.03
2	s:CreativeWork	411 298	425 224	0.03
3	s:Intangible	199 113	217 528	0.09
4	s:WebPage	157 473	167 276	0.06
5	s:Organization	152 437	175 588	0.15
6	s:Article	151 520	154 253	0.02
7	s:Blog	110 531	112 368	0.02
8	s:Place	110 011	131 634	0.20
9	s:StructuredValue	104 517	114 098	0.09
10	s:ContactPoint	101 298	110 074	0.09
11	s:PostalAddress	100 960	109 641	0.09
12	s:LocalBusiness	99 105	113 218	0.14
13	s:Product	91 168	92 714	0.02
14	s:BlogPosting	65 320	67 346	0.03
15	s:Offer	63 902	66 733	0.04
16	s:Rating	54 880	58 214	0.06
17	s:AggregateRating	50 475	53 593	0.06
18	s:Person	47 868	52 063	0.09
19	s:MediaObject	32 353	34 736	0.07
20	s:ImageObject	25 529	27 875	0.09
21	s:Event	20 275	20 471	0.01
22	s:Review	20 107	20 804	0.03
23	s:WebPageElement	14 055	40 552	1.89
24	s:ProfessionalService	10 113	10 199	0.01
25	s:GeoCoordinates	9 939	10 062	0.01
26	s:SiteNavigationElement	9 540	9 726	0.02
27	s:UserInteraction	9 192	9 206	0.00
28	s:UserComments	9 128	9 141	0.00
29	s:AutomotiveBusiness	9 007	11 872	0.32
30	s:WPFooter	8 440	9 046	0.07
31	s:WPHeader	7 879	8 490	0.08
32	s:AutoDealer	7 860	10 116	0.29
33	s:Recipe	7 578	7 648	0.01
34	s:VideoObject	7 419	8 284	0.12
35	s:WPSideBar	6 980	7 287	0.04
36	s:LodgingBusiness	5 677	5 816	0.02
37	s:Attorney	5 589	5 616	0.00
38	s:Hotel	4 722	4 785	0.01
39	s:Store	4 672	13 521	1.89
40	s:CollectionPage	3 888	3 903	0.00

Table 7.4: Top 40 most commonly deployed properties, ordered by the number of web sites using them within $S1$.

	Property	S1	S4	Δ %
1	s:name	518 759	552 235	0.06
2	s:image	254 176	255 368	0.00
3	s:url	241 804	270 089	0.12
4	s:description	237 610	240 355	0.01
5	s:address	104 615	106 985	0.02
6	s:addressLocality	97 544	98 674	0.01
7	s:streetAddress	97 298	98 268	0.01
8	s:telephone	91 204	93 236	0.02
9	s:postalCode	85 405	86 495	0.01
10	s:addressRegion	84 307	85 149	0.01
11	s:thumbnailUrl	73 952	74 723	0.01
12	s:author	71 204	71 476	0.00
13	s:datePublished	64 196	65 947	0.03
14	s:price	63 096	63 759	0.01
15	s:offers	62 978	64 087	0.02
16	s:articleBody	57 472	57 722	0.00
17	s:aggregateRating	52 471	53 475	0.02
18	s:ratingValue	51 618	51 821	0.00
19	s:mainContentOfPage	49 181	53 651	0.09
20	s:headline	46 059	52 112	0.13
21	s:availability	39 271	39 547	0.01
22	s:priceCurrency	33 241	34 133	0.03
23	s:blogPost	32 375	33 893	0.05
24	s:reviewCount	30 634	30 890	0.01
25	s:bestRating	30 233	30 390	0.01
26	s:logo	26 123	26 349	0.01
27	s:text	25 509	28 331	0.11
28	s:email	24 317	24 580	0.01
29	s:ratingCount	23 023	23 213	0.01
30	s:breadcrumb	22 865	22 922	0.00
31	s:faxNumber	19 403	19 665	0.01
32	s:addressCountry	17 817	17 942	0.01
33	s:keywords	15 994	16 063	0.00
34	s:creator	15 044	15 486	0.03
35	s:inLanguage	14 228	14 284	0.00
36	s:interactionCount	13 892	13 924	0.00
37	s:brand	13 199	14 786	0.12
38	s:contentURL	13 157	16 705	0.27
39	s:reviewRating	12 578	12 663	0.01
40	s:dateModified	12 379	12 412	0.00

effect of the different normalization steps of the proposed pipeline.[85] Applying a local syntactic duplicate removal reduces the total number of different entities (from $S1$ to $S2$) by almost 60%. Outstanding reduction rates can be observed for `s:Rating` and `s:AggregateRating` due to the small number of properties (e.g., `s:bestRating`, `s:ratingValue`, `s:worstRating`) and their limited capacity of different values (e.g., 1 to 5).[86] Apart from `s:Rating`, `s:Brand` and `s:Airport` are the classes with the largest degree of reduction. These classes are repeatedly used with `s:Product` and `s:Flight`, and there number is small compared to the number of entities of the classes they are used with (there are far less brands than products, and far less airports than flight connections).

Going further, we see that applying the heuristics and duplicate removal increase the number of entities again by 8%. Especially entities of class `s:MediaObject`, `s:ImageObject`, `s:WebPageElement` and `s:SiteNavigationElement` are often added or corrected by the heuristics. In the last step, applying a global syntactic duplicate removal, we end up with 40% of the number of entities we have in $S1$. Having a closer look at the last column of the table, we find that `s:JobPosting`, together with `s:AutomotiveBusiness` and `s:AutoRepair`, are the most invariant classes, where only less than 20% of the entities are detected as duplicates.

Table 7.7 lists the 20 most commonly used properties, based on the number of entities they are used by. Again, we report the difference Δ to the former step within the proposed pipeline as well as the difference Δ to $S1$.[87] Besides the already discussed case of ratings, the properties which are most drastically reduced are `s:addressLocality` and `s:addressRegion`, i.e., cities and states – entities that appear in many addresses, but have a rather small number of actual entities.

Overall, we can observe a similar behavior as for the classes. Applying the local syntactic duplicate removal in the first place reduces the overall number of entities by almost 60% which is reflected also within the properties that are used to describe those entities. The corrections of common mistakes using the heuristics have less influence on the number of properties than on the number of classes. Only for the properties `s:url` and `s:name`, we can find a large increase. This is mostly due to the inclusion of additional entities in cases where an object property has been represented by a literal value. The final step decreases the number of properties in the corpus again by around 8%.

7.3.2 Semantic Duplicate Detection

In the following, we selected 10 of the most frequently deployed classes and identified possible (pseudo-)key properties (compare Table 7.1). For those classes,

[85] Note that the total number of entities in this table is slightly lower than the numbers reported in Table 7.2, since in Table 7.5 and 7.6, we only count classes which are actually defined by the schema, meaning that typos and custom made classes are not covered.

[86] See Section 7.4.2 for a more detailed discussion.

[87] Unlike for classes, we omit to report the top 40 as the additionally gained insights are negligible.

Table 7.5: Most commonly deployed classes (Rank 1 to 20) by number of entities (in millions), including their subclasses, ordered by the number of entities within $S1$.

	Class	S1	S2	$\Delta_{1,2}$ %	S4	$\Delta_{2,4}$ %	$\Delta_{1,4}$ %	S5	$\Delta_{4,5}$ %	$\Delta_{1,5}$ %
1	`s:Thing`	1 459.5	594.4	−0.59	644.5	0.08	−0.56	589.3	−0.09	−0.60
2	`s:Intangible`	440.6	133.6	−0.70	137.0	0.03	−0.69	125.0	−0.09	−0.72
3	`s:CreativeWork`	344.0	172.0	−0.50	192.6	0.12	−0.44	175.9	−0.09	−0.49
4	`s:Product`	293.0	174.9	−0.40	175.5	0.00	−0.40	162.2	−0.08	−0.45
5	`s:Offer`	245.5	92.5	−0.62	93.3	0.01	−0.62	86.2	−0.08	−0.65
6	`s:Organization`	150.7	49.4	−0.67	53.6	0.08	−0.64	49.2	−0.08	−0.67
7	`s:Person`	115.4	28.5	−0.75	29.4	0.03	−0.75	26.2	−0.11	−0.77
8	`s:Place`	109.5	35.8	−0.67	36.5	0.02	−0.67	33.6	−0.08	−0.69
9	`s:Rating`	98.2	2.4	−0.98	2.7	0.12	−0.97	1.9	−0.31	−0.98
10	`s:Article`	90.6	47.4	−0.48	47.8	0.01	−0.47	42.5	−0.11	−0.53
11	`s:StructuredValue`	66.7	16.7	−0.75	17.3	0.04	−0.74	15.4	−0.11	−0.77
12	`s:MediaObject`	66.2	43.3	−0.35	55.3	0.28	−0.16	51.2	−0.08	−0.23
13	`s:WebPage`	65.3	25.9	−0.60	26.4	0.02	−0.60	24.2	−0.08	−0.63
14	`s:AggregateRating`	59.1	2.3	−0.96	2.5	0.09	−0.96	1.7	−0.32	−0.97
15	`s:ContactPoint`	50.6	12.1	−0.76	12.6	0.04	−0.75	11.2	−0.11	−0.78
16	`s:PostalAddress`	48.8	11.7	−0.76	12.4	0.06	−0.75	11.0	−0.11	−0.77
17	`s:Review`	42.6	13.3	−0.69	13.4	0.01	−0.68	12.5	−0.07	−0.71
18	`s:LocalBusiness`	41.8	21.6	−0.48	21.9	0.02	−0.48	20.3	−0.08	−0.52
19	`s:ImageObject`	35.4	19.3	−0.45	27.9	0.45	−0.21	25.7	−0.08	−0.27
20	`s:CivicStructure`	27.3	6.1	−0.78	6.1	0.00	−0.78	5.6	−0.08	−0.79

$\Delta_{a,b}$ refers to the difference between step a (Sa) and step b (Sb) within the proposed pipeline.

Table 7.6: Most commonly deployed classes (Rank 21 to 40) by number of entities (in millions), including their subclasses, ordered by the number of entities within $S1$.

	Class	S1	S2	$\Delta_{1,2}$ %	S4	$\Delta_{2,4}$ %	$\Delta_{1,4}$ %	S5	$\Delta_{4,5}$ %	$\Delta_{1,5}$ %
21	s:Airport	26.8	5.8	−0.78	5.8	0.00	−0.78	5.4	−0.08	−0.80
22	s:Event	24.0	13.1	−0.45	13.9	0.06	−0.42	12.9	−0.07	−0.46
23	s:NewsArticle	23.9	9.6	−0.60	9.8	0.02	−0.59	9.1	−0.08	−0.62
24	s:JobPosting	22.8	20.8	−0.09	20.8	0.00	−0.09	19.2	−0.08	−0.16
25	s:WebPageElement	19.7	5.2	−0.74	7.7	0.49	−0.61	7.1	−0.08	−0.64
26	s:UserInteraction	16.6	10.1	−0.39	10.1	0.00	−0.39	9.3	−0.07	−0.44
27	s:UserComments	15.8	9.3	−0.41	9.3	0.00	−0.41	8.6	−0.07	−0.46
28	s:MusicRecording	14.1	9.8	−0.31	10.2	0.04	−0.28	9.4	−0.08	−0.33
29	s:GeoCoordinates	14.1	4.0	−0.72	4.0	0.00	−0.72	3.5	−0.11	−0.75
30	s:VideoObject	12.2	8.9	−0.27	9.7	0.09	−0.21	9.0	−0.07	−0.27
31	s:BlogPosting	11.5	6.3	−0.45	6.4	0.01	−0.44	5.9	−0.08	−0.48
32	s:AggregateOffer	8.5	2.7	−0.68	2.7	0.00	−0.68	2.5	−0.08	−0.71
33	s:Residence	8.3	3.8	−0.54	3.9	0.01	−0.53	3.6	−0.07	−0.57
34	s:SiteNavigationElement	8.1	4.0	−0.50	4.9	0.21	−0.40	4.5	−0.08	−0.44
35	s:LodgingBusiness	7.6	2.8	−0.63	2.8	0.00	−0.63	2.6	−0.07	−0.66
36	s:Book	7.4	4.3	−0.42	4.3	0.00	−0.42	4.0	−0.07	−0.46
37	s:AutomotiveBusiness	7.2	6.2	−0.14	6.2	0.00	−0.14	5.7	−0.07	−0.20
38	s:AutoRepair	6.9	6.1	−0.12	6.1	0.00	−0.12	5.6	−0.07	−0.18
39	s:Hotel	6.3	1.9	−0.69	1.9	0.01	−0.69	1.8	−0.07	−0.71
40	s:Brand	5.4	1.1	−0.79	1.1	0.00	−0.79	1.1	−0.07	−0.80

$\Delta_{a,b}$ refers to the difference between step a (Sa) and step b (Sb) within the proposed pipeline.

Table 7.7: Top 20 most commonly used properties (in millions), ordered by the number of entities they are used by within $S1$.

	Property	S1	S2	$\Delta_{1,2}$ %	S4	$\Delta_{2,4}$ %	$\Delta_{1,4}$ %	S5	$\Delta_{4,5}$ %	$\Delta_{1,5}$ %
1	`s:name`	850.4	393.1	−0.54	417.2	0.06	−0.51	381.0	−0.09	−0.55
2	`s:url`	593.4	347.3	−0.41	390.0	0.12	−0.34	360.9	−0.07	−0.39
3	`s:image`	428.9	208.0	−0.51	208.6	0.00	−0.51	193.5	−0.07	−0.55
4	`s:offers`	242.8	177.0	−0.27	178.1	0.01	−0.27	164.6	−0.08	−0.32
5	`s:price`	214.7	71.3	−0.67	71.3	0.00	−0.67	64.8	−0.09	−0.70
6	`s:description`	202.9	128.7	−0.37	129.1	0.00	−0.36	119.5	−0.07	−0.41
7	`s:keywords`	143.7	131.7	−0.08	131.7	0.00	−0.08	121.8	−0.08	−0.15
8	`s:author`	110.8	57.9	−0.48	57.9	0.00	−0.48	53.7	−0.07	−0.52
9	`s:ratingValue`	91.1	3.9	−0.96	3.9	0.00	−0.96	2.8	−0.30	−0.97
10	`s:priceCurrency`	89.7	25.0	−0.72	25.1	0.00	−0.72	23.1	−0.08	−0.74
11	`s:datePublished`	80.2	42.6	−0.47	43.1	0.01	−0.46	40.0	−0.07	−0.50
12	`s:brand`	67.2	49.7	−0.26	49.9	0.00	−0.26	45.6	−0.08	−0.32
13	`s:address`	64.4	24.3	−0.62	24.3	0.00	−0.62	22.5	−0.08	−0.65
14	`s:addressLocality`	63.3	12.2	−0.81	12.4	0.01	−0.80	11.0	−0.11	−0.83
15	`s:aggregateRating`	61.2	32.0	−0.48	32.1	0.00	−0.48	29.8	−0.07	−0.51
16	`s:availability`	58.4	18.3	−0.69	18.3	0.00	−0.69	16.9	−0.07	−0.71
17	`s:bestRating`	54.8	2.0	−0.96	2.0	0.00	−0.96	1.6	−0.21	−0.97
18	`s:itemOffered`	54.0	33.9	−0.37	33.8	0.00	−0.37	31.5	−0.07	−0.42
19	`s:addressRegion`	47.9	9.7	−0.80	9.8	0.01	−0.80	8.7	−0.11	−0.82
20	`s:interactionCount`	47.6	29.0	−0.39	29.0	0.00	−0.39	26.9	−0.07	−0.43

$\Delta_{a,b}$ refers to the difference between step a (Sa) and step b (Sb) within the proposed pipeline.

Table 7.8 states the total number of entities E in $S5$, the number of entities making use of the identified (pseudo-)key property E_{pk} and the number of entities whose pk-value is not unique, meaning that they are duplicates $E_{pk,dup}$. As described in Section 7.2, from those sets, we can derive an estimation how many duplicates we can potentially find within $S5$. We can further calculate a more accurate number of different entities within the corpus. The last three columns in Table 7.8 show the number of entities, which we estimate to be duplicates, the estimated number of unique entities and the percentage differences compared to the total number of entities.

First, we can see that neither for `s:Offer` nor for `s:Person`, we could identify any entities making use of the particular pks. Second, in general, the probability of a duplicate ($P(dup)$), as well as the reduction rate (Δ), is highly class-specific. While only 6% of the entities of class `s:LocalBusiness` are estimated to be duplicates, this estimation is around 63% for `s:Product`. Regarding the final number of estimated *unique* entities in the corpus, we found that for `s:LocalBusiness`, `s:BlogPosting`, and `s:Organization`, the number of entities differs by less than 10% to the total number of entities. In contrast, the number of entities for `s:Product` and `s:WebPage` is reduced by 57% and 42% respectively.

7.4 Discussion

In this section, we discuss the major issues we have identified within the proposed method and our analysis in more depth.

7.4.1 Quality of Structural Duplicate Detection

In our pipeline, we use RDF graph equivalence between two entities to detect their equivalence. Meaning, that we consider two entities to be equal whenever they are described by the same RDF statements. Such an approach bares some flaws, as data providers do not need to annotate all information, which in some cases might lead to wrongly detected duplicates.

In order to get an estimation of the frequency of real duplicates versus misleading duplicates, we manually inspected 100 randomly selected pairs of identical subgraphs. The selected pairs contain subgraphs from the same but also from different web sites. Within those, only five entities were included which were annotated with the same information but do not refer to the same real-world object. These entities were either annotated with solely one property-value pair, or the values were left empty. From the remaining cases, 79 could be clearly marked as real duplicates which mainly result from annotations in headers and footers as well as from duplicated content in shops and overview pages. The remaining examples are hard to decide and solely use the class `s:WebPage`. This class should be used to annotate web pages in contrast to `s:WebSite`, which is used to annotate sites. But in all cases the `s:WebPage` is only annotated by the `s:url`, where the generic URL

Table 7.8: Estimation of the number of duplicate entities, based on semantic duplicate detection for selected classes (in thousands).

Class	#Entities in E	E_{pk}	$E_{pk,dup}$	$P(dup)$	# different pk-values in $E_{pk,dup}$	avg. #Entities w. same pk-value	est. #Entities dupl. in E	uniq. in E	Δ
s:Article	42 502.3	24 786.8	5 517.7	0.22	1 663.49	3.32	9 461.4	35 893.4	−0.16
s:Blog	2 384.2	388.0	79.7	0.21	11.73	6.79	489.6	1 966.7	−0.18
s:BlogPosting	5 906.6	2 682.2	297.7	0.11	108.10	2.75	655.6	5 489.0	−0.07
s:ImageObject	25 730.0	11 621.5	4 182.9	0.36	974.25	4.29	9 261.0	18 626.1	−0.28
s:LocalBusiness	20 271.0	7.4	0.4	0.06	0.12	3.72	1 170.2	19 415.3	−0.04
s:Offer	86 162.9	0.0	0.0	–	–	–	–	–	–
s:Organization	49 183.5	9.5	0.9	0.09	0.14	6.32	4 550.9	45 353.1	−0.08
s:Person	26 193.0	0.0	0.0	–	–	–	–	–	–
s:Product	162 167.7	978.0	502.3	0.51	88.48	5.68	83 280.3	93 558.5	−0.42
s:WebPage	24 195.2	12 948.8	8 103.2	0.63	719.96	11.26	15 141.0	10 399.5	−0.57

of the web site is used. This indicates that the detection of a duplicate is correct, but the used class is wrong for the indicated purpose.

7.4.2 Limitation of Duplicate Detection by RDF Graph Equivalence

While the used equivalence detection approach is straightforward, there are a few limitations and caveats which can be identified. First, we do not detect duplicates if the RDF graph describing one entity is not exactly the same, but a subgraph of that describing other entity. An example would be: two descriptions of two stores which are entirely the same, with the only difference being that one description also contains the opening hours, while the other does not. However, while our approach is deterministic w.r.t. eliminating duplicates, an approach also allowing subgraphs could potentially deliver different results depending on the execution order. For example, if for two entities, each one has a different subgraph contained in the other, the unified result would be different depending on which subgraph is merged first. For that reason, we did not apply an inclusion-based duplicate detection approach. A more comprehensive, yet more challenging, method could first check if one entity of a class is included in another. This could for example result from an overview page, where a certain number of entities is listed, but with only a reduced number of properties. The detail page then includes more information, capturing all information from the overview page. This method, however, does not only come at a higher computational cost, but also at the price of being non-deterministic in some cases. Second, many entities are underspecified. For example, if two products (e.g., shoes) are only described by the same name, brand, and price, we detect them as duplicated semantic annotations. However, the real-world object described by the semantic annotations may have different sizes and colors – but as they are not modeled in the semantic annotations, we cannot distinguish them in the duplicate detection process. That being said, the result of our process is a duplicate detection in the *extracted data*. However, it does not necessarily remove duplicates of *real-world objects*.

An extreme case of this is the `s:Rating` class. Usually, ratings have integer values between 1 and 5, in this extreme scenario, a duplicate detection on `s:Rating` objects that only have a rating value would result in only five individuals.[88]

7.4.3 Limitation of Heuristics with High Precision

As already stated before, in the proposed pipeline, we only apply heuristics for correcting schema violations with a precision of 100%. Meaning, we dismiss all heuristics which might lead (even only in rare cases) to wrong corrections. Therefore, some of our applied heuristics have a comparable low recall. Depending on the use cases, it might also be helpful to even apply heuristics, which are based

[88] It is a rather philosophical question whether two ratings with a value of 2, and without any further information such as a comment or details of the person who issued the rating, should be the same or not.

on observations, not always correct, but reflect reasonable approximations. In this chapter, we try to profile the dataspace of semantic annotations in a general and therefore omit the usage of less precise heuristics.

7.4.4 Selection of (Pseudo-)Key Properties

As we have already shown in the previous section, for a couple of classes, we have identified a set of properties whose values potentially could be used as identifiers. For some classes, e.g., `s:Product`, the properties such as `s:gtin13` are obvious and clearly defined to be used as keys. For other classes, such as `s:WebPage`, the property `s:url` might be the URL of the page or of the domain, which would need further pre-processing to be usable.

In addition, by inspecting the schema, we found some the identified pks to be of limited utility when applying them to the actual data. An example is the property `s:articleBody` for `s:Article` or `s:BlogPosting`. The general intention of that property is that its value includes the whole article or blog posting and therefore can be used to compare two entities. However, in some cases, pages annotate each paragraph or even each sentence with an instance of this property.[89] As the statements are unordered, the usage of such properties is difficult. In addition, to reduce the manual effort, it would also be possible to use automated methods to identify potential (pseudo-)key property candidates e.g., as described by [Hogan et al., 2010b]. Such methods, however, would need to be carefully tested and evaluated on the schema.org Microdata dataspace before they can be integrated into the pipeline.

7.5 Related Work

As we already have mentioned related work in the field of profiling in general and in particular for data retrieved from the Web, we will omit a repetitive mentioning in this chapter. We therefore focus on related work from the area of data quality and data cleaning.

A general study on the quality of the HTML code of web pages focusing on validation problems has been performed by [Chen et al., 2005]. They found that only 5% of all web pages are valid according to HTML standards, and analyzed the major problems leading to this invalidity.

A work summarizing trends in cleaning relational data, especially in the area of duplicate detection and consistency has been presented by [Ilyas and Chu, 2012]. The authors mainly focus on the two mentioned areas, where they define consistency as the *violations of integrity constraints* which is an extension to their former work [Chu et al., 2013]. Besides presenting techniques to detect the violations and duplicates, they also present ways in order to correct those errors. The presented methods are divided in methods which (1) only correct data, where a minimum

[89] Since schema.org does not define any cardinalities for properties, such violations can also not be detected automatically.

of cells should be changes as presented in the work by [Bohannon et al., 2005] and by [Kolahi and Lakshmanan, 2009]. Furthermore, they present (2) repairing algorithms which consider violations resulting from different types of integration constraints, which they summarize as *holistic* repairing [Fan et al., 2013] as well as [Fan et al., 2014]. The last (3) repairing technique presented assumes that the data itself is clean and only the constraints, or rules need to be changed. Several tools exists which support the user in cleaning data, either by providing more or less automatic cleaning or by requesting specific user feedback [Geerts et al., 2013, Dallachiesa et al., 2013, Chu et al., 2015].

In our work, we also focus on the detection and removal of duplicates but we do not focus on consistency to such an extent as realized by [Ilyas and Chu, 2012]. We detect and correct problems of range violations, but only on the base of object versus literal. Where an obvious next step is also the correction of violations of for example datatypes (string, number, date).

A review of the definition of the schema.org vocabulary and its mapping to RDF triples and OWL has been done by [Patel-Schneider, 2014]. This review uses a top-down strategy starting from the schema definition. However, it does not have a look at the data at all, which prevents the study from detecting possible influences on the outcome of data quality analysis.

Unfortunately, so far no other work exists which focuses on the analysis and cleansing of errors within semantic annotations. But, in the area of LOD, a larger body of work exists, dealing with this particular topic.

Linked Open Data [Bizer et al., 2009] and [Zaveri et al., 2015] have performed a study analyzing the quality of such data. While many of the metrics they applied for LOD are rather LOD-specific (such as the *presence* and *correctness* of dataset interlinks), some of the typical mistakes apply to both LOD and Microdata, mainly in the categories of *validity* and *consistency*. A survey which analyzes the general conformance of published LOD in terms of the so called *five-star Linked Data scheme* [Berners-Lee, 2006] has been performed by [Hogan et al., 2012]. Their work also compares the popularity of data providers using the *PageRank* towards the conformance to Linked Data guidelines. They found that popular pages often re-use vocabularies and support external linkage, while they do not tend to confirm the restrictions of usage of RDF features. [Harth et al., 2009] propose a method for ranking the results of RDF data retrieved from web sources, in order to overcome the negative effects from various web-specific peculiarities such as domain and structural variety, spam, and noise. They make use of the notion of a *naming authority* to combine the weight of an instance identifier and the authority, e.g., the data provider who assigns this identifier. Such a ranking method cannot directly be applied to the dataspace of Microdata, since in that dataspace, identifiers are used only scarcely [Meusel et al., 2015b]. Nevertheless, such an adapted approach

could later be used for Microdata in order to resolve conflicts during the data fusion process, where two or more participants provide different values for the same property.

One of the most related work, also aiming on the identification and removal of duplicates within the dataspace of schema.org Microdata, is by [Hogan et al., 2010a], who identifies four different categories of mistakes in LOD, i.e., *incomplete*, *incoherent*, *hijack*, and *inconsistent*. In the heuristics which we proposed before, are useful to overcome violations of the schema target especially on the aspects of *hijack* and *inconsistent*. We omit the aspect of incoherent and incomplete as those are difficult to identify because the definition of schema.org is rather relaxed. Similar to those papers, *Prolod++* [Abedjan et al., 2014], among others, can search for and identify typical modeling problems such as datatype properties with inconsistent data values (e.g., the representation of `release_date` of a movie as full date, month and year, and year only). The work by [Abedjan et al., 2012] even goes one step further. Using the deployment of LOD, their work aims to check whether properties are attached to the "right level" of the hierarchy or if certain properties should be redefined. The *LOD Laundromat* project by [Beek et al., 2014] provides cleaned versions of LOD datasets with syntax errors removed. This is comparable to our own previous work for Microdata [Meusel and Paulheim, 2015] as well as the methodology presented in this chapter.

7.6 Summary

In this chapter, we have focused on an approach for the topical profiling of deployed schema.org Microdata, which exploits different means for addressing data quality issues, arising from violations of the schema compliance as well as duplicate entries. Furthermore, we have shown that applying heuristics for correcting those issues can visibly change the outcome of the data profiling.

Impact on the Overall Profile We have presented a data profiling pipeline with integrated data cleaning methods. We have shown that for semantic annotations annotated using Microdata with schema.org extracted from web corpora, the initial data profiling results change drastically when applying duplicate detection on various levels as well as fixing obvious schema discrepancies using basic heuristics. These basic heuristics have a strong, increasing impact on the number of web sites which provide information about specific classes where the duplicate detection mechanism has a stronger, decreasing impact on the total number of entities which can be found within the data corpus.

Correction of Schema Violations Within this chapter, we applied basic heuristics in order to fix obvious errors in the deployment of the schema.org vocabulary. The heuristics focus on high precision, which lead for some of the heuristics to a low recall. Regarding the deployment of classes, the overall number of web

sites providing schema-compliant class information is increased by 3%, which is mainly due to the correction of typos. In particular the number of web sites describing organizations is increased by 15% and the number of web sites providing information about places by 20%. Both changes are mainly due to the heuristics handling object properties with formerly literal values. Going together with those corrections, the number of web sites making use of the property `s:name` or `s:url` increased by 6% and 12% respectively.

Structural Duplicate Removal With the presented pipeline, we found that the estimated number of web sites making use of a certain class can differ by up to 10% from a naive estimate. This means, that in some cases the number of web sites offering information about a class is 10% higher than it was before applying the pipeline. Regarding the number of entities within the corpus, using the initial number as baseline, we found that for some classes the number is decreased by over 50% using a simple duplicate detection mechanism.

Semantic Duplicate Removal Besides basic duplicate detection strategies, we have also analyzed the effect of semantic identity resolution, i.e., the identification of duplicates which are not RDF equivalents. To that end, we have selected a set of classes for which we could find (pseudo-)key properties, and estimated the effect of semantic duplicate removal. The result is highly class-specific, leading to an additional reduction by roughly 4% up to 57%, which has to be taken into account for the final data profile.

Chapter 8

Evolution of the Deployment of schema.org Microdata over Time

In the Chapter 6, we have analyzed the overall and semantic markup language-specific adoption of semantic annotations in the Web and examined the covered topics by the dataspace. We found, especially for the vocabulary schema.org together with markup language Microdata, a decent development within the last years, where the former chapter provided further insights into this particular dataspace and analyzed issues resulting from schema violations and duplicates.

This chapter focuses on the adoption and evolution of the deployment of schema.org over time in more detail. The official development of the definition of schema.org is managed by the W3C community group. The rather small number of participants of this group (in comparison to the number of schema.org data providers)[90] discusses change requests, additions and extensions to the definition of the schema.org vocabulary (which can be filed by anyone). As already mentioned in Chapter 2, the vocabulary has undergone more than 25 revisions since 2011, ranging from small typo fixes in vocabulary terms to the integration of entire new vocabularies.

At the same time, millions of web data providers use schema.org to mark up data on the Web. As shown in [Meusel and Paulheim, 2015] and Chapter 7, the actually deployed data can heavily deviate from the standard definitions. Frequent deviations include the usage of undefined classes and properties, as well as the usage of elements in a context in which they are not supposed to be used. Reasons for this can only be guessed but might be due to insufficient knowledge of the definitions or the deliberate decision to not stick to the definition, as it might be to complex or not appropriate.

In this chapter, we attempt an empirical, data-driven analysis of the interaction between those two groups. Such an empirical study is possible as we have snapshots from different points in time of the Web and thereby from the adoption of the dataspace. In addition, we can access the definition of schema.org which was valid

[90] Although anyone can join the group, the group has only 227 members, effective July 2017.

before and after the point in time of the snapshot. More specifically, we look at top-down and bottom-up processes. For the former, we analyze how fast and to which extent changes in the schema are adopted by data providers. For the latter, we examine how strongly changes in the schema are driven by undefined, yet frequent usages of schema elements. Wherever possible, we also try to find influences driven by the data consumers e.g., the tutorials provided by search engine companies such as Bing, Google, Yahoo!, and Yandex. Making use of this novel two-sided methodology to analyze the adoption and evolution of the deployment of schema.org over last four years, we reveal useful insights about the dataspace.

On the one hand side, those insights help to understand the state-of-the art deployment. On the other hand side, it allows the prognosis for the further development in this area. Both points are helpful and necessary for a later usage and integration of data from this dataspace into applications.

Therefore, the next section first describes the data corpus used in order to perform our empirical study. Section 8.2 specifies the research questions along with the proposed methodology to answer them. The findings for different aspects of the interaction of the two groups are presented in Section 8.3. The last section summarizes the findings.

The methodology described in the next sections as well as the results presented in Section 8.3 were already published in [Meusel et al., 2015a].

8.1 Research Data

As mentioned before, in order to perform our analysis, we need a representative snapshot of deployed semantic annotations using schema.org within the Web from different years, as well as the valid definition of the vocabulary before and after the point in time of the snapshot.

Therefore, we make use of the three different Microdata corpora from the different years as described in Chapter 4. As for this research, we focus on the adoption and development of the schema.org vocabulary and remove all other vocabularies (mainly data-vocabulary) from the corpora, similar to the process in the previous chapter. Although the vocabulary schema.org can potentially be used as well together with RDFa, we discovered that only around 0.1% of all web sites make use of this vocabulary together with RDFa, based on the results of Chapter 6.

Table 8.1 provides an overview of the size of the final corpora, we used within our analysis. As mentioned before, schema.org is promoted by the four world-wide largest search engine companies, Bing, Google, Yahoo!, and Yandex, and is maintained by an active user/developer group which discusses and maintains the schema.org schema definition. This community frequently creates whole new releases of the schema, where new classes and properties are introduced or domains and ranges of properties are changed. Also classes and properties are superseded by others or completely removed from the schema.

Table 8.1: Statistics of the filtered Microdata corpus, retrieved from the WDC project, containing only schema.org related data.

Corpus	Extraction Year	# Web Pages	# Web Sites	# RDF Statements
C_{2012}	2012	19 281 189	29 413	232 687 529
C_{2013}	2013	217 751 199	399 139	6 411 276 458
C_{2014}	2014	232 279 437	731 573	6 778 845 785

Table 8.2: Overview of the different sets of changes between the selected schema.org releases. We separate changes introducing new concepts and properties (new) and removing existing concepts and properties (dep).

	Release		# Classes		# Propert.		# Domain/Range		#
Δ	from	to	new	dep	new	dep	new	dep	supersession
S_1	`0.91`	`1.0c`	233	0	387	2	23	2	0
$S_{1'}$	`0.95`	`1.0c`	140	0	161	0	34	1	0
S_2	`1.0c`	`1.91`	62	0	190	1	119	9	32
$S_{2'}$	`1.0f`	`1.91`	36	1	108	1	32	5	32
S_3	`1.91`	`1.93`	27	0	76	1	68	69	3
$S_{3'}$	`1.92`	`1.93`	2	0	15	1	22	5	0

We have extracted the RDF schema of the releases before and after the three crawls, i.e., release `0.91`, `0.93`, `1.0c`, `1.0f`, `1.91`, `1.92` and `1.93`, using the Internet Archive[91] for older releases, and the schema.org GitHub repository[92] for the newer ones. Figure 8.1 depicts the temporal order of the three used web corpora and the analyzed schema.org changes between the selected release versions.

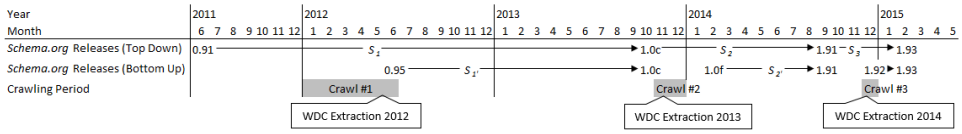


Figure 8.1: Timeline of schema.org release dates and web corpora dates.

Table 8.2 shows the number of newly introduced classes and properties for each of the selected releases in comparison to the previous one, as well as the number of domain/range changes, deprecations, and supersessions.

In this table, S_i denotes the changes of the schema between the time when crawl i was started, and the time when crawl i was finished. These change sets are used to analyze top-down processes, i.e., adoptions of changes in the schema.

In contrast, S_i' denotes the changes of the standard between the *end* of crawl i and the beginning of crawl $i + 1$. These change sets are used to analyze bottom-up processes, i.e., influences of the deployed data on the standard.

[91] `http://web.archive.org/`
[92] `https://github.com/schemaorg`

8.2 Research Questions and Methodology

Within this section, we define the research questions and introduce the methodology which is later used to answer those questions.

Generally, as explained in the Chapter 9, due to potential bias in the crawl all our measures are based on the information aggregated by web site. For defining the measures, we use the following notation conventions: our corpora are denoted with C_{2012}, C_{2013}, and C_{2014}, as explicated above. For each corpus C_i, t_i denotes the time at which it was collected, and $\#WS_i$ denotes the total number of web sites in the corpus deploying schema.org Microdata. Furthermore, for a triple pattern T, we define $\#WS_i(T)$ as the total number of web sites in a corpus which use the triple pattern T at least once.[93]

To quantify the usage of a class c and a property p, we define

$$\#WS_i(c) := \#WS_i(\texttt{?x rdf:type c}) \tag{8.1}$$

$$\#WS_i(p) := \#WS_i(\texttt{?x p ?y}) \tag{8.2}$$

as the total number of usages of types and properties aggregated by web site.

8.2.1 Top-down Processes

Top-down processes are "schema first" processes, meaning that the standard changes and the data providers follow the standard. Here, we analyze how changes in the standard are reflected in the data *after* the change has been defined.

More specifically, we investigate in the following three aspects:

- Adoption of new classes and properties
- Implementation of deprecations
- Implementation of domain/range changes

For measuring the adoption of a new schema element s (i.e., a class or a property), we determine the *normalized usage increase (nui)* of that element as

$$nui_{ij}(s) := \frac{\#WS_i(s)}{\#WS_j(s)+1} / \frac{\#WS_i}{\#WS_j} \quad (i > j). \tag{8.3}$$

The nominator of the overall fraction denotes the increase in the usage of s between the corpora C_i and C_j, whereas the denominator denotes the general increase of semantic annotations using schema.org Microdata contained in the Web. In order to avoid division by zero for elements that have not been used previously, we use $\#WS_j(s)+1$ as a denominator instead of $\#WS_j(s)$.

The usage of a normalized measure is steered by the raw data which is used for our analysis. The underlying web crawls – based on nature of web crawls – do not

[93] We use the notion of triple patterns as defined in the W3C SPARQL standard [Prud'Hommeaux et al., 2008].

include the same sets of web pages and web sites and do also not include the same number of crawled pages. By this, even if the total number of adopting sites could be larger, the relative amount could be smaller as in the crawl before. These facts forbid the usage of non-normalized scores such as the simple differences between the total number of pages adopting a particular class.

For a new schema element s added to the standard between two released t_i and t_{i+1}, we say that it has been successfully adopted, if there is a $i > j$ so that $nui_{ij}(s) \geq 1.05$, i.e., the increase of the usage of the element is significantly larger than the overall increase in schema.org Microdata. Likewise, for deprecated elements, we say that the deprecation has been successfully adopted, if $nui_{ij}(s) \leq 0.95$, i.e., the usage of the element has significantly decreased.

The rationale for the normalization is that, assuming there are no other influencing factors, it can be expected that the usage increase of an element increases proportionally with the overall increase of the corpus. Only if the usage increase of an element significantly *exceeds* this expected increase, we can say that there is a measurable impact of the change in the standard.

For domain and range changes, we have to distinguish between classes being *added* to the domain/range definition, and classes being removed. Due to the disjunctive interpretation of domain and range definitions in the schema of schema.org [Patel-Schneider, 2014], the former *broadens* the possible usages of a property, while the latter *restricts* the possible usages, i.e., the latter can lead to formerly legal definitions to become illegal.

For measuring the adoption of domain changes of a property p and a domain d, we count triple patterns of the type

```
?x p ?y .  ?x rdf:type d',
```
(8.4)

where `d'` is a subtype of `d`, or `d` itself. For range changes with a range `r`, we count triple patterns of the type

```
?x p ?y .  ?y rdf:type r',
```
(8.5)

where `r'` is a subtype of `r`, or `r` itself. For such a patterns, we define $nui_{ij}(p)$ as in (8.3).

As for new classes and properties, we say that an addition to a domain/range definition is adopted if the corresponding $nui_{ij}(p) \geq 1.05$. We say that a removal from a domain/range definition is adopted if the corresponding $nui_{ij}(p) \leq 0.95$.

8.2.2 Bottom-up Processes

Bottom-up processes are "data first" processes, meaning that the standard is adapted to its actual implementations and deployments. Here, we analyze how changes in the standard are reflected in the data *before* the change has been defined.

More specifically, we have a look at the following aspects:

- The usage of (undefined) classes and properties before they were officially announced.
- The adoption of schema.org's extension mechanism[94] to define new classes.
- The usage of properties with subjects and objects not defined in their range.

To measure whether there are such bottom-up processes, we hypothesize that elements that are already used by a larger number of data providers *prior to announcement* are likelier to be included in the standard. As a measure for testing this hypothesis, we use *receiver operator characteristics*, i.e., ROC curves [Fawcett, 2006].

ROC curves are often used in measuring the performance of predictors, such as machine learning trained classifiers. Here, we measure if the number of web sites which deploy a specific schema element is a good predictor for that element to become officially added to the standard.

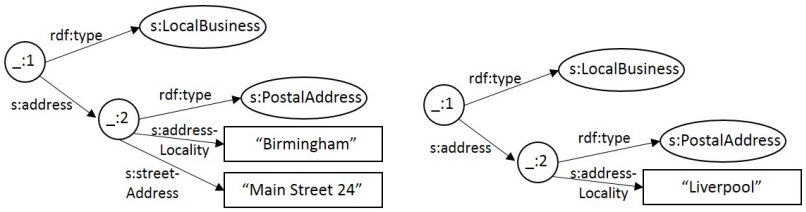
The ROC curves are constructed as follows: Given two corpora C_i and C_{i+1}, let A_{i+1} denote the set of schema elements that have been *added* to the standard between t_i and t_{i+1}. Furthermore, let S_i be the list of undefined schema elements (according to the standard at time t_i) used in C_i, ordered by $\#WS_i(s)$. Then, we mark each element in S_i as a *true positive* if it is also contained in A_{i+1}, as a *false positive* otherwise. Given the ordered list, we graph the true positive rate against the false positive rate, and measure the *area under the curve* (AUC), which is normalized to a $[0, 1]$ range. We build individual ROC curves for classes and properties.

If $AUC = 0.5$, then there is no influence of the usage of an element on the probability of it being included into the standard. For $AUC > 0.5$, there is a positive influence (i.e., more frequently deployed elements are likelier to be included into the standard later), if $AUC < 0.5$, there is even a negative influence.

Likewise, we analyze whether the usage of the extension mechanism has an influence on the standardization. For example, the class `s:Artwork` has been newly introduced in a recent schema.org release, being a subclass of `s:CreativeWork`. Before that official introduction, it could have been used via the extension mechanism by defining the class `s:CreativeWork/Artwork`, which then recognized as a user-defined subclass of `s:CreativeWork`. Like for inofficially used elements, we compare the list of schema elements defined in C_i using the extension mechanism to the corresponding set A_{i+1} of new schema elements in the standard at t_{i+1}, and compute ROC curves both for classes and properties.

To measure whether domain and range changes are influenced by the actual usage of data, we look specifically at domain and range definitions that have become *broader*. To that end, we look at all domain and range usages in a corpus C_i according to Equation 8.4 and Equation 8.5 which are not defined in the standard at t_i. Again, we sort them by $\#WS_i(p)$ and mark all domain/range definitions that

[94] `http://schema.org/docs/extension.html`



(a) Document 1 (b) Document 2

Figure 8.2: Two RDF graphs retrieved from example documents describing a `s:LocalBusiness` entity. The second graph omits the description of a street.

have been added to the standard at t_{i+1} as true positives, the rest as false positives. The resulting ROC curves show if there is a tendency to add domain and range definitions based on the deployed usage.

8.2.3 Overall Convergence of Vocabulary Usage

The third question we raise is the overall convergence or divergence of schema.org Microdata. Specifically, we want to know if the diversity of representing particular entities – such as an address – has increased or decreased over time. Convergence is a plausible scenario due to the increased availability of tutorials and best practices, e.g., for *Google's Rich Snippets* [Google Inc., 2015], or adaption to the consumers of Microdata, such as optimization w.r.t. search engine rankings. Divergence is also possible due to the larger number of adopters, all of which come from different domains and backgrounds with specific requirements.

To quantify convergence and diversity, we adapt a normalized entropy measure [Shannon, 2001]. Since the RDF data extracted from a web page forms a cycle-free RDF graph with a defined set of roots [Hickson et al., 2014], we first describe the *vocabulary term usage* of the page as the ordered enumeration of all paths from any root to any leaf. For the paths, we extract the types and properties, but omit blank node identifiers and literal values. To enforce an ordering, we use a simple lexicographic ordering (similar to the approach used for duplicate detection in Chapter 7). Exemplary, the two documents shown in Figure 8.2 would be described by the following enumerations:

$$\begin{aligned} S_1 = \{ \; & \texttt{s:LocalBusiness} \rightarrow \texttt{s:address/s:PostalAddress}, \\ & \texttt{s:LocalBusiness} \rightarrow \texttt{s:address} \rightarrow \texttt{s:addressLocality}, \\ & \texttt{s:LocalBusiness} \rightarrow \texttt{s:address} \rightarrow \texttt{s:streetAddress} \} \end{aligned} \quad (8.6)$$

and

$$\begin{aligned} S_2 = \{ \; & \texttt{s:LocalBusiness} \rightarrow \texttt{s:address/s:PostalAddress}, \\ & \texttt{s:LocalBusiness} \rightarrow \texttt{s:address} \rightarrow \texttt{s:addressLocality} \} \end{aligned} \quad (8.7)$$

The set of those enumerations is now treated as a sequence of symbols, where each enumeration element (i.e., each path) is understood as a symbol. Thus, we can compute the total entropy for the set of the two example documents as

$$H = \sum_{i=1}^{n} -p(path_i)log(p(path_i)). \tag{8.8}$$

In the above example, the total entropy would be 1.918, using the dual logarithm. We normalize this by dividing by the product of the total number of paths N in a corpus, and

$$H_{max} = log(n) \tag{8.9}$$

where n is the total number of *different* paths, to account for effects of different corpus sizes (i.e., we use a *normalized entropy rate*):

$$H_{norm} = \frac{H}{H_{max} \cdot N}. \tag{8.10}$$

In the above example, this would lead to a normalized entropy rate of 0.192.

If we now assume that at a point in time, the second document would also add a `s:streetAddress`, i.e., the two documents would become more alike, the normalized entropy rate would drop to 0.167. Thus, we can observe that the description of entities of the class `s:LocalBusiness` has become more uniform.

We compute an overall normalized entropy, as well as normalized entropies per class defined in schema.org.

8.2.4 Influence of Data Consumers

As stated earlier, one major incentive to use semantic annotations in the HTML pages is, among other things, an improved display on the search result page. Google, the most widely used search engine[95], calls those improved displays *Rich Snippets* and supports web site provides within their Google Developer Tools pages for structured data with How-Tos and examples, as discussed above. In particular Google promotes the seven different topical domains `s:Products`, `s:Recipes`, `s:Reviews`, `s:Events`, `s:SoftwareApps`, `s:Videos`, and `s:Articles`, and explicitly states which properties are required for being properly displayed within their Rich Snippets. In our analysis, we will look at the measures for those classes in isolation, where appropriate.

In addition, one could assume that the introduction of easy-to-use code snippets (e.g., examples) how to implement a certain description/markup language within HTML can boost the deployment. However, such examples have been available on the schema.org web site even before the first corpus we use, so we do not expect any significant findings from the availability of such examples.

[95] `http://gs.statcounter.com/#search_engine-ww-monthly-200807-201503`

8.3 Empirical Findings

In this section, we describe and analyze the empirical findings for top-down and bottom-up processes, as well as the overall convergence of semantic annotations embedded using Microdata together with schema.org.

8.3.1 Top-down Processes

Based on the timeline in Figure 8.1, for the following analysis we consider the set of changes S_1, including the changes between the releases `0.91` and `1.0c`, and the set of changes S_2 including the changes between the releases `1.0c` to `1.91`. We make use of the three mentioned corpora to calculate the normalized usage increase (nui) between two corpora i and j for a schema element s based on the number of web sites making use of this particular element.

Adoption of New Classes and Properties

Overall, we observe that from the 233 new classes introduced with S_1, 113 could *not be observed at all* in any of our corpora until 2014. Likewise, regarding the 389 new properties in S_1, 109 could not be observed until 2014. Within the change set S_2, 23 out of 62 newly introduced classes and 109 out of 190 newly introduced properties were not used at all in the 2014 corpus.

These findings show that the adoption of new classes and properties in general is happening slowly, and that there are certain parts of the schema which are barely used at all in the public Web. This is also reflected in the average and median nui-values reported in Table 8.3. In particular, the low median values show that the vast majority of newly introduced classes and properties is not significantly adopted.[96]

Table 8.3: Median and average nui-values of classes and properties.

	Classes		Properties	
Change Set	S_1	S_2	S_1	S_2
Median	0.00	0.55	0.12	0.00
Average	0.47	8.04	2.01	6.63

By manually inspecting the lists of non-adopted classes and properties, we could identify three particular domains. For elements introduced in S_1, we could not find any evidence for parts of the objects (1) from the medical domain (e.g., `s:Nerve` or `s:Vein`), as well as (2) for many specific subclasses of `s:Action`. The first is an effect of integrating an existing large-scale, multi-purpose vocabulary for a

[96] A median of 1.05 would confirm that half of the classes/properties are significantly adopted.

domain – in this case, the medical domain – into the schema[97], where not all parts of that vocabulary are equally useful for marking up the content of a web page.

The cause for the latter effect may be a blind spot of our corpora, because actions are basically designed for e-mail markup, not web page markup.[98] For elements introduced in S_2, the main domain of non-adopted elements are related to booking actions – here, again, the semantic markup is likely to be used in confirmation e-mails and form-based interaction with the deep web, both of which are not included in the data foundation.

Table 8.4 lists the 19 significant deployed classes of the change sets S_1 and S_2, which are at least deployed by five web sites in the 2014 corpus based their nui-value. Although we have found a large fraction of action-related and medical-related classes beyond those not being adopted at all (see above), two classes from those domains, i.e., `s:SearchAction` and `s:MedicalIndication`, are listed in the table. Within S_2, we find a large fraction of classes related to the broadcasting domain, as well as services. In addition, the FAQ-related classes are present like `s:Question` and `s:Answer` as schema.org has been adopted by major question-and-answer sites such as *Stack Overflow*[99].

Furthermore, Table 8.4 lists the seven classes which are promoted by Google's Developer Tools in order to mark content for their *Rich Snippets*. Although the nui-value is below 0.95, the absolute number of web sites using those classes is still growing, but not as fast as the overall deployment of schema.org in Microdata. Especially the classes `s:VideoObject` (64%), `s:Product` (58%), and `s:Review` (53%) have strongly increased in the total number of web sites deploying the classes.

Regarding the properties introduced in S_1 and S_2, we found 13 and 56, respectively being significantly deployed within the 2014 corpus. Table 8.5 and Table 8.6 lists the together 42 significant deployed properties of both change sets based on their nui-value, which are deployed by at least five web sites. We added the possible domains for all of the properties in order to allow a straightforward grouping by topical domain.

A first observation, which we could already draw from the significant deployed classes (see above), is that a large fraction of those significantly deployed properties are only used by a small number of web sites, especially for S_1. Regarding S_2, we see a stronger deployment by number of web sites. This is remarkable, since time is apparently not a crucial factor in adoption, i.e., elements that have been present in the standard for a longer time are not necessarily adopted more widely. Similar to the classes, most of the properties have domains of the groups: `s:Action`, `s:MedicalEntity`, `s:CreativeWork`, `s:ContactPoint`, `s:Organization`, `s:Service`, `s:Question` and `s:Events`.

In order to gather additional insights and to identify groups of classes which define significantly properties, we calculated, based on the properties within S_1 and

[97] `http://blog.schema.org/2012/06/health-and-medical-vocabulary-for.html`

[98] `https://developers.google.com/gmail/markup/`

[99] `http://stackoverflow.com/`

Table 8.4: List of the 19 significant deployed classes of S_1 and S_2 between 2013 to 2014 being at least used by five web sites in 2014 and the 7 classes directly promoted by Google Developer Tool web page for Rich Snippet integration at the end of the table.

		# Web Sites		
Δ	Class	2013	2014	nui
S_1	s:SearchAction	1	11	3.00
S_1	s:MedicalIndication	1	5	1.36
S_1	s:BusinessFunction	2	7	1.27
S_2	s:WebSite	2	1 648	299.71
S_2	s:Car	0	76	41.46
S_2	s:QAPage	0	60	32.74
S_2	s:Answer	0	41	22.37
S_2	s:Question	1	47	12.82
S_2	s:PublicationIssue	0	21	11.46
S_2	s:Vehicle	0	21	11.46
S_2	s:Periodical	0	16	8.73
S_2	s:BroadcastEvent	0	14	7.64
S_2	s:BroadcastService	0	14	7.64
S_2	s:Episode	1	17	4.64
S_2	s:Service	18	147	4.22
S_2	s:EmailMessage	4	35	3.82
S_2	s:ServiceChannel	0	5	2.73
S_2	s:Airline	0	5	2.73
S_2	s:RadioEpisode	2	7	1.27
	s:Product	56 537	89 683	0.87
	s:Recipe	6 025	7 593	0.69
	s:Review	13 143	20 115	0.84
	s:Event	8 253	10 105	0.67
	s:SoftwareApplication	1 809	2 091	0.63
	s:VideoObject	4 516	7 424	0.90
	s:Article	65 864	88 569	0.73

Table 8.5: List of the 6 significant deployed properties of S_1 between 2013 to 2014, being deployed at least by five web sites in 2014.

			# Web Sites		
Δ	Property	Domains (Excerpt)	2013	2014	nui
S_1	s:result	Action	2	10	1.82
S_1	s:agent	Organization	1	6	1.64
S_1	s:endTime	Action	2	7	1.27
S_1	s:object	Action	3	9	1.23
S_1	s:codeValue	MedicalCode	4	11	1.20
S_1	s:medicineSystem	MedicalEntity	3	8	1.09

Table 8.6: List of the 36 significant deployed properties of S_2 between 2013 to 2014, being deployed at least by five web sites in 2014.

			# Web Sites		
Δ	Property	Domains (Excerpt)	2013	2014	nui
S_2	s:potentialAction	Thing	0	783	427.20
S_2	s:target	Action	0	783	427.20
S_2	s:commentCount	CreativeWork	0	98	53.47
S_2	s:hasMap	Place	0	54	29.46
S_2	s:contactOption	ContactPoint	1	101	27.55
S_2	s:doorTime	Event	1	80	21.82
S_2	s:pagination	Article	0	32	17.46
S_2	s:department	Organization	7	256	17.46
S_2	s:acceptedAnswer	Question	0	29	15.82
S_2	s:position	CreativeWork, ListItem	0	27	14.73
S_2	s:subOrganization	Organization	4	121	13.20
S_2	s:suggestedAnswer	Question	0	23	12.55
S_2	s:partOfSeries	Episode, Season	0	22	12.00
S_2	s:organizer	Event	2	65	11.82
S_2	s:areaServed	ContactPoint	0	21	11.46
S_2	s:answerCount	Question	0	18	9.82
S_2	s:productSupported	ContactPoint	1	34	9.28
S_2	s:upvoteCount	Anser, Comment, Question	0	17	9.28
S_2	s:serviceArea	Service	2	48	8.73
S_2	s:serviceType	Service	4	73	7.97
S_2	s:audienceType	Audience	0	13	7.09
S_2	s:accessibilityFeature	CreativeWork	0	12	6.55
S_2	s:issueNumber	PublicationIssue	0	12	6.55
S_2	s:availableLanguage	ContactPoint, ServiceChannel	0	11	6.00
S_2	s:mapType	Map	0	11	6.00
S_2	s:publishedOn	PublicationEvent	0	10	5.46
S_2	s:produces	Service	1	19	5.18
S_2	s:numberOfSeasons	*Series	0	9	4.91
S_2	s:directors	Episode, Movie	0	7	3.82
S_2	s:hasPart	CreativeWork	0	6	3.27
S_2	s:eventStatus	Event	2	17	3.09
S_2	s:hoursAvailable	ContactPoint	3	22	3.00
S_2	s:reservationFor	Reservation	0	5	2.73
S_2	s:publication	Clip, Episode, MediaObject	4	12	1.31
S_2	s:issn	Periodical	3	8	1.09
S_2	s:license	CreativeWork	17	36	1.09

Table 8.7: Excerpt of classes ordered by the calculated average nui based on properties from S_1 and S_2 for the 2013 and 2014 corpus.

Rank	Class	Avg. nui
1	`s:Action`	48.61
... other `s:Action`-subclasses		
64	`s:PostalAddress`	11.46
65	`s:ContactPoint`	11.46
66	`s:Service`	5.14
66	`s:Taxi`	5.14
68	`s:Question`	4.94
69	`s:Event`	4.28
... other `s:Event`-subclasses		
92	`s:TVSeason`	4.03
... other TV-related-subclasses		
... other mixed classes		
126	`s:Places`	3.41
... other `s:Place`-subclasses		
... other mixed classes		
181	`s:Organization`	1.62
... other `s:Organization`-subclasses		
280	`s:MedicalEntity`	0.82
... other `s:MedicalEntity`-subclasses		
... other classes		

S_2 for the comparison of 2013 and 2014 the average nui-values for all possible classes. For this calculation, we exclude all properties which are inherited from the class `s:Thing`, to reduce the noise resulting from too general properties.

The average values for the selected classes are displayed in Table 8.7. The table ranks all the classes we have identified earlier and all of them – except for `s:MedicalEntity` – have an average nui-value above the significance level.

Implementation of Deprecations

With the changes of S_1, no deprecations were introduced. Within S_2 one property became completely deprecated, but it was not used in any of the three corpora. Beside the complete deprecation of one property, 32 became superseded by others. Out of those 32, the usage of superseded properties significantly decreased for 29 of those, except for the three properties `s:map`,`s:maps` and `s:musicGroupMember`, where `s:map` is still significantly used in 2014. Major users of the `s:map` properties are e.g., the web sites of hotels like `marriott.com`, travel sites, and blogs like `travelpod.com`.

From the substitutes for the superseded properties which should be used after the changes of S_2, we could find nine to be adopted significantly within the 2014 corpus, listed in Table 8.8. For the remaining 23, there is no significant adoption for the substitutes.

Table 8.8: Significantly used substitutes of superseded properties of S_2 within the 2014 corpus. In all those cases, the property supersedes the respective property named by the plural form, i.e., `s:blogPost` supersedes `s:blogPosts`.

	# Web Sites		
Property	2013	2014	nui
`s:blogPost`	5 445	33 946	3.40
`s:employee`	214	745	1.90
`s:member`	254	871	1.87
`s:sibling`	3	9	1.64
`s:event`	149	369	1.35
`s:award`	102	235	1.26
`s:contactPoint`	331	726	1.20
`s:season`	42	85	1.10
`s:photo`	1 004	1 962	1.07

Implementation of Domain/Range Changes

Based on our observations, none of the range changes of properties, which were introduced within S_1, is significantly deployed in any of the three corpora. From the 18 introduced domain changes in S_1, six are adopted by a significant amount of web sites in the later corpora. The adoptions for those changes, which are deployed by at least five web sites in 2014, are listed in Table 8.9. Four are directly related to the *product* domain.

From the 12 significantly deployed domain changes (out of 87), Table 8.9 lists the eight which are used by at least five web sites in 2014. In addition, the table also includes the seven adopted range changes (out of 20) included in S_2. A large proportion of those adoptions can be assigned to the *broadcasting* domain. That domain was introduced into the schema.org vocabulary based on discussions and influence with BBC and EBU.[100]

8.3.2 Bottom-up Processes

In this section, we report on the numbers of classes, properties and other changes which are actually adopted by web pages before they became official within the schema definition of schema.org. In particular, we inspect the changes made starting from release `0.95` for the first corpus ($S_{1'}$, S_2, and S_3), and from release `1.0f` for the second corpus ($S_{2'}$ and S_3) and from release `1.93` to the current release for the last corpus ($S_{3'}$). We are aware of the fact that before a change is officially announced, there are ongoing discussions and proposals (which are all public), which also could affect the earlier adoption of non-official classes, and we will take this into account when drawing any conclusions.

[100] `http://blog.schema.org/2013/12/schemaorg-for-tv-and-radio-markup.html`

Table 8.9: List of domain/range changes significantly adopted and at least deployed by 5 web sites in 2014. $(+)$ indicates a new range/domain, $(-)$ the removal of a range or domain.

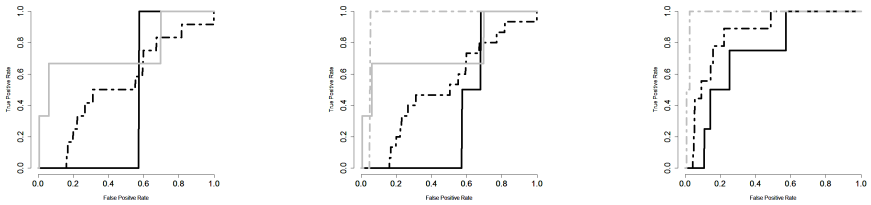
Δ	Change	# Web Sites 2013	2014	nui
Domain				
S_1	s:Product/width $(+)$	99	318	1.73
S_1	s:Product/itemCondition $(+)$	360	1 187	1.79
S_1	s:Drug/manufacturer $(+)$	13	32	1.25
S_1	s:PriceSpecification/priceCurrency $(+)$	100	215	1.16
S_1	s:Product/height $(+)$	85	299	1.90
S_2	s:Event/typicalAgeRange $(+)$	0	6	3.27
S_2	s:Organization/memberOf $(+)$	1	5	1.36
S_2	s:TVEpisode/episodeNumber $(-)$	75	106	0.76
S_2	s:Thing/alternateName $(+)$	1	14	3.82
S_2	s:Service/provider $(+)$	2	55	10.00
S_2	s:WebPage/isPartOf $(-)$	43	68	0.84
S_2	s:RadioSeries/episode $(+)$	0	5	2.73
S_2	s:Episode/actor $(+)$	1	7	1.91
Range				
S_2	s:comment|s:Comment $(+)$	44	172	2.13
S_2	s:seasons|s:TVSeason $(-)$	9	8	0.48
S_2	s:episodes|s:TVEpisode $(-)$	11	14	0.69
S_2	s:partOfSeason|s:TVSeason $(-)$	15	22	0.80
S_2	s:isPartOf|s:CollectionPage $(-)$	18	30	0.91
S_2	s:episode|s:TVEpisode $(-)$	56	78	0.76
S_2	s:image|s:ImageObject $(+)$	101	264	1.43

Usage of Classes and Properties before Official Announcement

Regarding the usage of (undefined) classes and properties within the deployed data before they were officially included in the standard, we can report in general a rather small pre-announcement deployment. From the changes of S_1', we identified only one class and 13 properties being already deployed in the corpus of 2012. The most deployed properties were `s:value` and `s:color` which were both used by four different web sites.

Analyzing the influence of the deployment in 2012 and 2013 for the changes until release `1.91`, we found that within the first corpus only the class `s:Service` was deployed by one web site and three other properties were already present. Within the corpus of 2013, we found four classes and eight properties being deployed. Those mainly belong to the domain of flights, where the class `s:Flight` was deployed by six different web sites together with their properties `s:iataCode`, `s:arrivalAirport`, and `s:departureAirport`. Those are not big airlines, but copies of one and the same meta-flight booking portal: `aviagid.com.ua`.

Regarding the influence of the deployed classes and properties for the `1.93` release, we found that the class `s:Game` was already used in 2012 by six web sites, and by 18 web sites in 2013 before it was officially released. We also found three and six properties, respectively being deployed in 2012 and 2013, with the property `s:currency` being used by 24 web sites in 2012 and already 551 in 2013



(a) 2012/2013 Bottom Up Dataset Comparison

(b) 2012/2014 Bottom Up Dataset Comparison

(c) 2013/2014 Bottom Up Dataset Comparison

Figure 8.3: ROC for each dataset comparison for classes (black line), properties (black dotted line), domain changes (grey line), and range changes (grey dotted line)

is most outstanding. Within the corpus of 2014, we could identify the property `s:material` being already used by six web sites before the official release.

As described in Section 8.2.2, we draw the ROC curves for the three corpora comparisons and calculate the corresponding AUC values. Figure 8.3 shows the different curves for the three comparisons for classes (black line) and properties (black dotted line). As stated above, for the classes, we could only identify one and two classes, respectively which are used before the official release, which explains the angular curves. For the properties, we could find more adoptions, but the curve also follows more or less the diagonal.

Table 8.10 shows the calculated AUC values for the comparisons based on the ROC curves of Figure 8.3. The values for classes and properties for the comparison between 2012 and 2013, 2012 and 2014, respectively show a more or less random distribution, where the comparison of 2013 and 2014 shows a stronger trend towards an influence of pre-official usage of classes and properties. Summarizing the influence, based on the average in the last column of this table, shows a minor trend overall.

Table 8.10: AUC values for bottom up adoption of classes, properties, and domain and range changes between the different datasets.

	2012/2013	2012/2014	2013/2014	Avg.
Classes	0.4272	0.3739	0.7305	0.5105
Properties	0.5369	0.5292	0.8547	0.6403
Domain Changes	0.7449	0.7449	–	0.7449
Range Changes	–	0.9498	0.9827	0.9662

Adoption of schema.org's Extension Mechanism

When looking for new classes and properties being used via the extension mechanism before their official introduction, we found only three class extensions in the 2012 corpus (`s:*/Service`, `s:*/Vehicle`, and `s:*/WebApplication`), being

used by maximum of two web sites. Furthermore, ten properties are introduced using this mechanism, with `s:*/softwareVersion` being used most frequent (by five web sites).

For the 2013 corpus, three properties were used with the extension mechanism and become later official. But the usage is always less than four web sites. Class-wise, we again find `s:*/Vehicle` being deployed using the extension mechanism by nine web sites, and seven further classes.

In 2014, we can report one class and five properties being introduced using the extension mechanism. Outstanding, again, is `s:*/currency`, which was used by ten web sites.

Overall, regarding the extension mechanism, we cannot report any significant influence on the newly introduced properties and classes. In general, the mechanism is not widely adopted, and we can observe that data providers are more likely to introduce classes and properties directly without using the official extension mechanism.

Usage of Properties with Subjects and Objects Outside their Defined Domain and Range

In addition to classes and properties, we analyzed the pre-official usage of domains and ranges with properties, where the domain/range was not defined yet at the point in time when the corpus was crawled. In other words, we look at properties being used in a different *context* than the one they were intended to be used.

Overall, we found that six domain/range changes (four domain and two range changes) can be detected within the crawled data before they become official. Especially the range changes `s:comment` with its new range `s:Comment` and `s:image` with its new range `s:ImageObject` are already used by over 40 and 100 web sites, respectively in the 2013 corpus. A prominent example for using a property with a new domain is `s:Organziation/brand`, which was already present on 255 web sites in 2012 although it was not official released.

We again draw the ROC curves as described in Section 8.2 and display the different curves for domain and range changes within Figure 8.3. From those curves and the corresponding AUC values, depicted in Table 8.10, we can observe that at least for domain and range changes, the schema evolution is driven by the real world usage to a certain extent, as the AUC values of those changes are significantly larger than 0.5.

8.3.3 Overall Convergence of Vocabulary Usage

To complete the picture of the evolution of deployed semantic annotations over time, we inspected the development of the heterogeneity of the usage of the different class definitions and also of the global dataspace.

As described in Section 8.2.3, we use an entropy-based measure for measuring heterogeneity. From an overall point of view, we find that the global normalized

entropy rate, as defined in Equation 8.10, and hence the heterogeneity, decreased from 2012 ($2.34e^{-09}$) to 2014 ($9.42e^{-11}$) by around $2\,400\%$, i.e., we can observe a strong homogenization of the data representations.

Regarding the class-wise entropy and its development from 2012 to 2014, we have a closer look at the 57 most deployed classes (classes which we could find on more than 1 000 web sites in the 2014 corpus)[101], the entropy decreases for 56. Only the entropy for the class `s:VideoObject` increased by around 18%. Comparing only the class-wise entropy for the 2013 and the 2014 corpus, we can report that 37 increase in homogeneity and 17 decrease.

Table 8.11 list those 37 classes for which we found a decrease of their class-specific entropy from 2013 to 2014.

The classes listed here can be grouped in four different categories:

1. Classes describing **web sites, their elements and structure** like `s:WebSite`, `s:ImageGallery`, `s:Blog`, `s:WPSidebar`, `s:WPHeader`. The increase in homogeneity of such kind of classes can be explained by the increasing adoption of schema.org within Content Management Systems.[102]
2. **Services and facilities**, like `s:Florist`, `s:AutoDealer`, `s:Hotel`, `s:Restaurant`, and `s:Store`, mostly belonging to the class of `s:LocalBusiness`. Those classes are mainly deployed by *Yellow Pages* web sites.
3. **Products and offers**, like `s:Product`, `s:ItemList`, and `s:Offer`. Here the promotion of Google's Developer Tools and Rich Snippets could be a possible driver.
4. **Ratings and reviews**, which can be found in all of the three categories above: `s:Rating` and `s:Review`.

At the other end of the spectrum, Table 8.12 lists the 17 classes with a decrease in homogeneity between 2013 and 2014. Here, we can observe two high-level classes, i.e., `s:LocalBusiness` and `s:Event`, for which a larger number of specific subclasses have been introduced in later releases of schema.org, so that instances of those classes have become richer (and more diverse) in their descriptions.

Another group of classes with increasing heterogeneity are classes describing locations, like, e.g., `s:PostalAddress`, `s:Place`, `s:GeoCoordinates`. As shown in [Meusel and Paulheim, 2015], those classes were mostly used erroneously in the 2013 corpus, thus, the change to a more "correct" representation might lead to this decrease (i.e., the descriptions get more heterogeneous since correct and incorrect representations are used side by side).

[101] For this experiment, we focus only on classes which were already deployed in 2012.

[102] For example the CMS Drupal (starting with Version 7) automatically annotates the generated pages within their system using schema.org classes and properties: `https://www.drupal.org/project/schemaorg`.

Table 8.11: List of classes with an increase of homogeneity from 2013 to 2014. Column 2 states the class where the third column reports the change of the entropy. 100% in this column means, that the current (2014) entropy is only half of the entropy in 2013. Asterisks mark the promoted classes by Google's Developer Tools.

Rank	Class	Increase of Homogeneity (%)	# Web Sites in WDC 2014
1	s:WebSite	> 1 000.00	1 650
2	s:Thing	> 1 000.00	79 967
3	s:SiteNavigationElement	> 1 000.00	9 540
4	s:ImageGallery	> 1 000.00	1 679
5	s:RealEstateAgent	929.61	2 133
6	s:Florist	883.75	1 571
7	s:ItemList	582.59	1 697
8	s:Blog	422.45	110 531
9	s:IndividualProduct	378.50	1 403
10	s:WebPage	302.52	148 710
11	s:UserComments	251.87	9 128
12	s:AutoDealer	201.03	7 860
13	s:OpeningHoursSpecification	155.96	1 163
14	s:Book	116.27	1 674
15	s:Dentist	114.03	2 410
16	s:SearchResultsPage	104.54	1 123
*17	s:Product	98.04	89 579
18	s:Movie	81.87	2 171
*19	s:Recipe	70.64	7 578
20	s:Corporation	65.44	1 900
21	s:CollectionPage	63.61	2 127
22	s:Offer	49.94	62 828
23	s:NutritionInformation	49.30	1 274
24	s:Brand	49.06	2 486
25	s:BlogPosting	41.36	65 320
26	s:ItemPage	30.82	3 455
27	s:WPSideBar	29.77	6 980
*28	s:VideoObject	28.06	7 419
29	s:Hotel	24.66	4 722
*30	s:SoftwareApplication	17.45	2 087
31	s:Rating	15.34	12 183
*32	s:Review	14.08	20 107
33	s:JobPosting	8.64	2 838
34	s:Store	5.94	1 819
35	s:Restaurant	4.26	2 524
*36	s:Article	1.16	88 164
37	s:WPHeader	0.56	7 879

Table 8.12: List of classes with a decrease of homogeneity from 2013 to 2014. Asterisks mark the promoted classes by Google's Developer Tools.

Rank	Class	Increase of Homogeneity in %	# Web Sites in 2014
1	s:ProfessionalService	88.73	1 197
2	s:LocalBusiness	74.36	62 131
3	s:NewsArticle	73.98	2 514
4	s:ProfilePage	66.49	3 377
5	s:PostalAddress	65.14	100 960
*6	s:Event	57.57	10 091
7	s:MusicGroup	43.84	2 010
8	s:Place	30.87	9 912
9	s:AggregateOffer	30.69	2 038
10	s:Person	28.02	47 868
11	s:ApartmentComplex	26.60	1 921
12	s:ImageObject	20.67	25 529
13	s:CreativeWork	12.97	6 226
14	s:WPFooter	12.73	8 440
15	s:ContactPoint	8.42	1 034
16	s:Organization	4.61	52 658
17	s:GeoCoordinates	2.60	9 939

8.4 Related Work

As we already have mentioned related work in the field of profiling in general and in particular for data retrieved from the Web, we will omit a repetitive mentioning in this chapter.

The combination of semantic markup languages and vocabularies to include semantic annotations in HTML pages is not the first, and will not be the last standard which is proposed to the Web. From time to time new standards are introduced (as HTML5) or standards are changed and they get or do not get adopted. Therefore, it is not surprising that a larger body of work exists examining the adoption of different technical standards as well as the role of standardization bodies in this process [Chiao et al., 2007]. Example studies investigating the factors that drive the adoption of electronic data interchange (EDI) standards include [Chau and Hui, 2001, Chen, 2003, Yee-Loong Chong and Ooi, 2008]. Studies that focus on the adoption of web service technologies include [Chen, 2005, Ciganek et al., 2006]. A more closely related work, as presented in this chapter, is on the adoption of specific vocabularies for publishing semantic annotations on the Web, described in [Ashraf et al., 2011] and [Kowalczuk et al., 2014]. Both investigate the adoption of the *GoodRelations* vocabulary for representing e-commerce data. A study of the adoption of the Web Ontology Language (OWL) is presented by [Glimm et al., 2012]. Our work distinguishes itself from these work by focusing on a different vocabulary and analyzing the diffusion of the vocabulary over a longer time span.

8.5 Summary

In this chapter, we have shown how the availability of data deployed on the Web using a given standard, and the standard itself, allows for a new kind of empirical analysis of standard adoption, which is completely data-driven. The findings from our quantitative analysis are manifold:

Diversity of Deployed Classes By far not all elements introduced in the definition of schema.org are actually deployed in the popular part of the Web – in fact, about half of the defined elements could not be observed in any of the corpora. On the other hand, deprecations in the standard are most often adopted quite well. This indicates for data consumers, that whenever a class is removed, related semantic annotations will not be found anymore in the Web at some point in time.

Usage of Properties within a new Context We have also shown that bottom-up processes influence the evolution of the schema. In particular, the usage of defined properties in new contexts (i.e., with other domains and ranges than defined in the schema) often leads to corresponding changes in the schema.

Usage of the schema.org Extension Mechanism The intended way of introducing new classes and properties, i.e., the usage of schema.org extension mechanism, is much less used than the (unintended) direct deployment of a new class or property. Especially with respect to the fact that the schema.org community just recently promoted a new way to extend the existing schema, those findings are interesting. In the future, it needs to be observed if the newly introduced extension mechanism is used by data providers more frequently, which our findings do not indicate.

Homogeneity of schema.org Entities We can observe that the homogeneity increases, e.g., (a) when there is a global player consuming the corresponding data, such as Google with its Rich Snippets for search results enrichment, or (b) by adoption of schema.org in widely deployed content management systems. This fact is especially interesting for data consumers, as they can expect more and more data providers deploying semantic annotations in a similar way. Consequently, the number of erroneous semantic annotations will decrease over time.

Influence of Data Consumers We have identified, that the data consumers, such as Google and likely also Facebook influence the way on how data providers annotate the data. Although the provided vocabulary definitions are rich, from a large fraction of data providers only the bare necessities in order to be consumed is used. This indicates, that whenever more detailed described semantic annotations or semantic annotations for a certain topic are needed, data providers need to be offered a direct benefit.

In conclusion, we generate useful insights for data consumers, as well as data providers of semantic annotations with the presented data-driven analysis. Using snapshots of the Web at different points in time, we are able to observe processes in the adoption of a standard, as well as its evolution. Such observed processes, as well as the identification of key drivers and obstacles in standard adoption are also useful insights for the adoption of other standards, such as Linked Open Data [Paulheim, 2015].

Part III

Use Case-Specific Profiling

Chapter 9

Use Case-specific Utility Analysis of schema.org Data

Within the previous part of the thesis, the applied profiling methods and the resulting findings were mainly independent from any specific use case or application. Although we used profiling in order to detect and remove duplicates and correct schema violations in terms of data cleansing, this application is rather generic, as its results are helpful for a large fraction of other, subsequent use cases.

As already mentioned by [Naumann, 2014], in most cases data profiling is driven and steered by a concrete application or use case. Meaning, the profiling and the resulting findings can help to answer questions about the utility of the dataspace to support the range of functions of a concrete application.

Therefore, within this chapter, we move the scope of the analysis towards the utility of the dataspace of semantic annotations in HTML pages using Microformats, RDFa, and Microdata for concrete use cases. In particular, we analyze to which extend required information for the use cases of a *marketplace*, a *news aggregator*, and a *travel portal* are provided by the schema.org Microdata dataspace. Answering questions about the utility of semantic annotations for a particular use case is an important step in the direction of the subsequent integration of the data in related applications.

In the following section, we first outline the three use cases in detail, stating the information requirements which needs to fulfill by semantic annotations, to be considered as useful for the particular use cases. Section 9.2 summarizes related work from the area of use case-specific profiling of semantic annotations. Subsequently we explain the methodology which was applied to examine the utility for the three use cases. Afterward, we describe the data used to perform the empirical study. In Section 9.4 the findings of the study are outlined and critically discussed with respect to the analysis of the utility of the dataspace for the three use cases in the last section.

9.1 Use Cases

This section describes the three before mentioned use cases in more detail. We use the use cases and the related information requirements to showcase the utility of semantic annotations from the schema.org Microdata dataspace. The main question, arising for all the three use cases, is to which extend the defined information need can be satisfied by the information provided through semantic annotations embedded in use case-related web sites.

9.1.1 Marketplace

A marketplace, in the classic sense, is a space where different (re-)sellers compete with each other to sell their product. Especially in the Web, marketplaces and related applications are and will become more and more relevant, also from an economic perspective. Based on the study of [eMarketer, 2014], the digital buyer penetration continuously increased over the last years. Based on predictions of the trend from 2015 [eMarketer, 2015], this trend will continue at least till 2019. The study states, that the share of 7.4% of retail done via the internet will increase till 2019 to over 12.8%. This trends is also underlined by the strong adoption of the shopping-related classes like `s:Product` and `s:Offer`, as shown in Chapter 6. Exemplary, well-known, actual web marketplaces are hosted by the companies *Ebay* and *Amazon*. Here, (re-)sellers can offer their products and customers can directly compare the potentially different offers for the same or similar product(s) and can further buy the items through the marketplace. Other shapes of web marketplaces are price comparators like the one hosted by *Idealo*[103]. In comparison to the marketplaces provided by two before mentioned companies, the price comparators allow customers only to compare the prices of products, where the actual proceeding is done by the individual (re-)sellers. No matter, if the products are only displayed or additional services like the processing is also provided by the marketplace, for the core functionality, a set of information is required to be provided by the different resellers.. Therefore, for the use case of a marketplace, we are interested in the number of web sites providing products and product-related information. Furthermore, in order to allow customers to compare prices of similar offers, we require from the different web sites to provide the name, description, a picture and the price of the offered products. To improve the comparability of the products, as well as the customer experience, if available, information about the availability and condition as well as ratings or reviews are requested. In addition, a categorization of the described products is beneficial in order to allow a category-based filtering in the marketplace.

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

9.1.2 News Aggregator

A common example of a news aggregator is *Google News* where from different web sites the most recent or interesting news are presented based on the users personal interest. Similar to the marketplace, the news are gathered from different news providers (web sites or APIs), integrated by the news aggregator and displayed to the user. In order to provide an satisfactory user experience for each news item, we require as minimum the title, parts or the complete text of the article and a publishing date. Furthermore, a related image can help to improve the visibility of the article. Information about the author as well as about the copyright are potentially helpful for the use case to avoid legal issues.

9.1.3 Travel Portal

A travel portal, such as the one provided by *Tripadvisor*, *Kayak*, and *Trivago*, enable visitors to find and compare accommodations, as well as flights and/or other services required when traveling (such as car rentals). Within our use case of a travel portal, we restrict ourselves to information about accommodations (hotels, hostels, and so on). As the availability of such offers is usually very high, an important fact, which is also especially taken into account by the service provided by *Tripadvisor*, are reviews and ratings. Therefore, we define as the information we require for this use case the name of the hotel, and a URL to the hotel web site. We further need information about the location of the hotel to enable location-based functionalities like search and navigation. Furthermore, we require ratings and reviews. In addition, we would need to know if an accommodation is available within a certain time period, and the corresponding price. Based on the information requirements, we consider this use case as the most complex one.

9.2 Related Work

Within the application area of traveling, including hotels, hostels and other accommodations, [Toma et al., 2014] has analyzed how semantic annotations (e.g., embedding in HTML pages using Microdata together with schema.org) can improve the visibility of travel-related web sites and web pages in the Web. The authors emphasize that by using schema.org to provide semantic annotations for hotels the number of visits of the page increases on average by 8%. Although this number sounds low, in comparison to the relatively small effort needed to include semantic markup languages in HTML pages, the benefits are huge.

In [Stavrakantonakis et al., 2013], the authors analyzed the influence and possibilities of information and communication technologies within the Web 2.0 and Web 3.0, especially for the online presence of hotels in Austria. They show that although new techniques with cheep setup costs exists, they are not yet widely used. They identified a gap in the Austrian tourist service industry, respectively the industry providing the online presence for the hotels. The work by [Kärle et al., 2016]

deeper examined the adoption of schema.org and Microdata by travel-related sites in Austria, investigating further in the direction of the identified gap.

An analysis of the deployment of classes and properties by the most popular online shops was performed in one of our former works [Meusel et al., 2015b]. Here, we used a rather small number of online shopping web sites (32) and distinguished them into *producer*, *merchant*, and *marketplace*. We found that almost all web sites of the three groups annotate the provided data with at least the *name*, an *image*, the *description* as well as an *offer* containing the *price*. They rarely annotate the *availability* as well as the *price currency*.

A work, focusing on the usage of markup languages in the area of video platforms, is presented in [Kutuzov and Ionov, 2014]. The authors conclude that the most used semantic markup languages is Microdata together with schema.org. Their research is limited to Russian platforms and therefore the results might not be representative for other areas of the Web.

Besides the researches mentioned before, the search engine companies Bing, Google, Yahoo!, and Yandex, make use of semantic annotations for their applications [Guha et al., 2015], e.g., *Google's Rich Snippets*. Although it is known, that semantic annotations are used, no information are available about the internal integration of those information. The companies do not provide any studies about the spread of use case-specific semantic annotations, or how they are integrated into their application. Further it is also unknown if other sources of information are combined to improve the data quality and density.

9.3 Research Data and Methodology

This section briefly describes the used research data as well as the methodology which is applied to measure the utility of the dataspace with respect to the requirements of the use cases.

9.3.1 Research Data

In order to conduct our study of the utility of semantic annotations for the named use cases, in the following, we make use of the cleaned dataspace of schema.org Microdata, presented in Chapter 7. In particular, we make use of the corpus which is referred to as *S4*. As recap, the contained semantic annotations originate from the Microdata corpus of late 2014, and are only described by the schema.org vocabulary. Furthermore, from this corpus we removed duplicated entities within a web site and corrected schema violations, using the heuristics described in Section 7.2.2.

We want to analyze the utility of semantic annotations provided by use case-specific web sites to satisfy the information need of the three different use cases. Therefore, in a first step, we need to identify web sites related to the three use cases. We use the web site categorization provided by *Alexa*, a service which was founded in 1996 and is now owned by the *Amazon.com Company*. The service

Table 9.1: Number of different web sites gathered from Alexa, aggregated by the first-level category, including travel, ordered descending by the number of web sites.

Category	# Web Sites	Category	# Web Sites
Business	149 346	Health	26 944
Society	60 129	Science	24 251
Shopping	58 164	Reference	14 299
Recreation	50 776	Games	9 989
– Travel	4 818		
Arts	50 477	Home	7 379
Computers	43 540	News	4 056
Sports	39 499	Kids and Teens	3 028

delivers insights and analytics for web site owners. Thereunder, they also estimate the traffic of web sites and rank them based on this estimation. In addition, *Alexa* also provides a topical categorization of web sites. This categorization includes 15 topical categories for the first level such as *arts*, *games*, *shopping*, and *sports* to name just a few.[104] Using the *Alexa Web Information Service* (AWIS)[105], we retrieved for each of the 15 categories and their direct subcategories overall 903 307 entries. As entries do not necessarily need to be a web site but can also be a specific page or sub-domain, in a second step, we normalized retrieved entries and calculated for each entry the corresponding web site. This step results in 568 446 different web sites, where we further removed entries which do not belong to one unique category. We ended up with a set of 541 877 uniquely categorized web sites, where 535 530 also are categorized with a unique subcategory. The number of web sites per category is shown in Table 9.1.

We selected web sites from the *Shopping* category for our marketplace use case, web sites from *News* for the news aggregator use case, and web sites from the category *Travel*, which is a sub-category of *Recreation* for the use case of the travel portal.

9.3.2 Methodology

Chapter 6 presented the profiling of the adoption of semantic annotations and the discovery of topics within this dataspace in general. In contrast, within this chapter the studies focus on the profiling of use case-specific requirements.

First, the semantic annotations provided by the use case-specific web sites are analyzed based on the classes of the described entities. Further, the co-occurrence

[104] The categorization can be found under: `http://www.alexa.com/topsites/category`. We omit the categories *World* and *Regional*, as they represent a geographic categorization other than a topical.

[105] In particular, we used the `CategoryBrowse` and `CategoryListings` API actions, described within the AWS documentation: `http://docs.aws.amazon.com/AlexaWebInfoService/latest/`.

of different classes is inspected. The co-occurrence of classes on the same web sites reveals insights in order to discover usage patterns in semantic annotations. Such insights help, especially for use cases like the travel portal, to detect if information about hotels and reviews are provided frequently together from the same web sites or if they are more frequently provided in isolation. We make use of the *FP-Growth* algorithm (described in [Han and Pei, 2000]) to calculate frequent patterns within the embedded classes. Similarly, [Kowalczuk et al., 2014] extracted common usage patterns of the *GoodRelation* schema within the RDFa dataspace.

Subsequent, after the analysis of co-occurring classes, we investigate in the properties which are annotated in the relevant semantic annotations. We compare the provided information with the defined information need of each use case, as already discussed in Section 9.1.

As the subject of the studies is the analysis of the capability of the schema.org Microdata dataspace to satisfy use case-specific information requirements in general, we do not analyze the actual property values provided by the semantic annotations. Meaning, in this chapter, we only analyze if web sites at least make use of properties related to the identified information needs of the use cases, but do not analyze what specific values are provided for the different properties, and if they might be helpful.

9.4 Empirical Findings

This section presents the empirical findings for the three use case-specific groups of web site, based on the methodology presented before. In addition, in a first step, we perform a similar analysis of the dataspace stretched by all web sites making use of Microdata together with schema.org. This allows a comparison of the results of the use case-specific analysis in contrast to the overall profile.

9.4.1 Use Case-Independent Analysis

Inspecting the schema.org Microdata dataspace, we find $1\,800$ different classes and over $21\,000$ different properties being used by at least one web site. Regarding the number of classes and properties used by multiple web sites, the diversity decreases. Table 9.2 underlines this observation and presents the number of different classes and properties used by at least two, five and ten different web sites. Inspecting those dismissed classes, we identified them as web site-specific creations or typos, which cannot be easily fixed by the heuristics described in Chapter 7.

Class and Property Adoption per Site In addition to the total number of different classes and properties used in the corpus, we also analyzed the number of different classes and properties deployed per web site. Figure 9.1 depicts this distribution in an cumulative manner. We found that web sites deploy on average 2 different classes (2.03) and make use of 4 to 5 different properties (4.64). Having a look at the two extremes of the distribution, we find that at maximum the same

Table 9.2: Number of different classes deployed by at least 1, 2, 5 and 10 different web sites.

	# Different Classes	# Different Properties
1 web site	1 800	21 814
2 web sites	900	2 744
5 web sites	517	1 287
10 web sites	420	893

site makes use of 108 different schema.org classes. Those web sites are descriptive blogs about the schema.org vocabulary. Furthermore, we find that all sites use at least one class. But the percentage of web sites making use of more than one class, rapidly decreases, and we only find 47% of the sites using multiple classes. Regarding the properties, we discovered over 10% of the sites making no use of any schema.org related property, other than the `rdfs:type` property. Inspecting some of those sites, we find that those web sites solely state the class in the first HTML element, and do not provide any other semantic annotations in the following HTML. Going further, we discovered one web site, namely `empik.com`, deploying over 11 thousand different properties. This Polish shopping web site annotates their products with a set of various `s:Product`-specific properties. Besides, in order to identify a products, they introduce a identifier specific property, instead of using the `s:productID` property. This way of schema.org adoption results in this large number of different properties, which are never used by any other site.

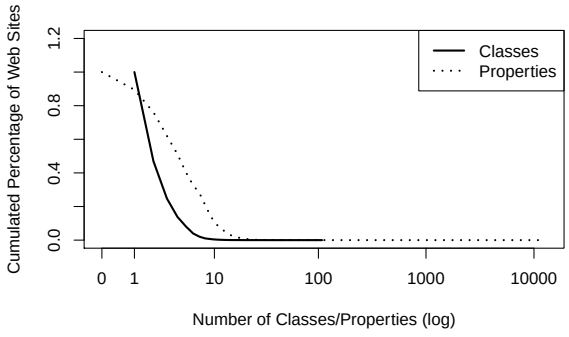


Figure 9.1: Accumulated percentage of web sites deploying different classes and properties per page.

Summarizing, we see that in general the web sites making use of Microdata together with schema.org provide only semantic annotations about one or two different kinds (classes) of entities. In addition, those information are rather shallow described, due to the small number of properties. This issue is further examined in [Petrovski et al., 2014].

Table 9.3: List of the 20 most visited shopping-related sites making use of schema.org with Microdata within CC 2014 based on Alexa.

Rank	Web Sites	Rank	Web Sites
1	bestbuy.com	11	wayfair.com
2	ikea.com	12	walgreens.com
3	macys.com	13	6pm.com
4	kohls.com	14	sephora.com
5	homedepot.com	15	zappos.com
6	nordstrom.com	16	gamestop.com
7	gap.com	17	bodybuilding.com
8	toysrus.com	18	samsclub.com
9	overstock.com	19	cvs.com
10	bhphotovideo.com	20	cabelas.com

9.4.2 Marketplace

From the list of *Shopping* web sites, retrieved by the Alexa classification (see Section 9.3), we identified 5 451 web sites providing semantic annotations contained in our schema.org Microdata corpus. Those web sites make on average use of 3 different classes per site (3.35) where the median is also 3. They also deploy on average around 8 different properties per web site (7.77), where the median is 7. In comparison to the overall average usage of different class and property, use case-related sites employ 50% more different classes and around double the number of properties, indicating a part of the dataspace which is annotated more extensively.

Top Sites Table 9.3 lists the top 20 most visited web sites based on the (visitor) ranking of Alexa. The products sold by those sites range from clothing (e.g., `macys.com` and `6pm.com`) over electronics (e.g., `bhphotovideo.com`) over interior fittings (e.g., `ikea.com`) to sporting equipment (e.g., `bodybuilding.com`).

Frequent Classes Table 9.4 lists the 20 most commonly used schema.org classes by use case-related sites. We observed that around 70% of those sites annotate their products using `s:Product`. We also find over 50% of the those sites making use of the class `s:Offer`. In addition, the product-related classes `s:ItemAvailability` and `s:AggregateRating` are deploy by 36% and 27% of the web sites, respectively. We also find classes which are not directly related to shopping. 11% of the sites make use of the class `s:WebPage`.

Frequently Co-occurring Classes In order to gain further insights, we calculate frequent itemsets of the classes embedded by the use case-related sites. In a first step, we are interested in the co-occurrence of other classes together with `s:Product`. Table 9.5 lists the 10 most common frequent itemsets for this category. The support is

Table 9.4: List of the 20 most common deployed schema.org classes by shopping-related sites.

	Class	Web Sites #	Web Sites %
1	s:Product	3 806	0.70
2	s:Offer	2 825	0.52
3	s:ItemAvailability	1 955	0.36
4	s:AggregateRating	1 450	0.27
5	s:Thing	1 186	0.22
6	s:Organization	1 157	0.21
7	s:WebPage	575	0.11
8	s:Review	468	0.09
9	s:OfferItemCondition	444	0.08
10	s:Store	404	0.07
11	s:PostalAddress	401	0.07
12	s:Rating	387	0.07
13	s:Article	340	0.06
14	s:LocalBusiness	273	0.05
15	s:Person	272	0.05
16	s:VideoObject	218	0.04
17	s:Blog	199	0.04
18	s:BlogPosting	135	0.02
19	s:Duration	109	0.02
20	s:Brand	101	0.02

calculated based on the overall number of sites in this subpart of the data. We found that in most cases, when a site makes use of the class `s:Offer` (52%) it is used together with `s:Product` (50.6%). The same holds for `s:ItemAvailability` and `s:AggregateRating`. In addition, we found that 14% of the sites make use of all four classes and provide rich information about the products they sell. Examples are the web sites `bestbuy.com` and `bodybuilding.com`.

From the web sites which do not make use of the `s:Product` class, we found that 27% make use of `s:WebPage` in order to annotate their pages. We also found 14% of such sites making use of the `s:Article` or `s:PostalAddress` class. As in a first step, we assumed that those sites are either wrongly annotated and do not belong to the domain of shopping, we manually inspected a random sample of them. We found, that all of them can be assigned to the domain of shopping, as all of them offer products. Some of them, especially those which annotate their pages using `s:WebPage` provide a mixture of information about a certain topic and a small shop, which offers topic-related products.

Table 9.5: List of the 10 most common frequent itemsets including `s:Product` deployed by shopping-related sites.

Support	Itemset
.506	`s:Product` `s:Offer`
.351	`s:Product` `s:ItemAvailability`
.346	`s:Product` `s:Offer` `s:ItemAvailability`
.252	`s:Product` `s:AggregateRating`
.196	`s:Product` `s:Offer` `s:AggregateRating`
.179	`s:Product` `s:Organization`
.166	`s:Product` `s:Offer` `s:Organization`
.161	`s:Product` `s:Thing`
.149	`s:Product` `s:ItemAvailability` `s:AggregateRating`
.147	`s:Product` `s:Offer` `s:ItemAvailability` `s:AggregateRating`

Product Information In the following, we focus on web sites making use of the classes `s:Product` and `s:Offer` in their semantic annotations. In particular, we want to analyze the richness of provided information and explore if the provided information meet the requirements of our use case. Table 9.6 lists the top 20 most common used properties which are used by web sites, annotating entities using the class `s:Product` and/or `s:Offer`. The proportion stated in column 4, is based the number of web sites making use of the `s:Product` and/or `s:Offer` class in total. Besides the object properties `s:offers`, `s:availability`, and `s:aggregateRating`, we find over 90% of the web sites annotate the products with a name, and over 70 provide a description, a price, and an image together with the product. Unfortunately, only 34% provide a specification of the price currency, together with the price which need to be taken into account, when gathering and comparing the prices for the use case of a marketplace. Furthermore, in the second half of the top most frequent properties, we find that less than 20% of the web sites provide information about the brand of a product, the item condition as well as more specific rating information (e.g., the best and worst rating). Within the list of the 20 most commonly used properties, we cannot find the property `s:category`, which is only used by less than 3% of the web sites.

9.4.3 News Aggregator

The second use case, which we are interested in, is the one of a news aggregator. Similar to before, we analyze first the general adoption of classes and properties by web sites related to news. We use the web sites categorized by this class by Alexa and identified 705 web sites of this category in our schema.org Microdata corpus. Those sites make on average use of 4 different classes per site (3.94) where the median is 3. In terms of properties, news-related sites deploy on average 9 different properties (9.42) where the median is 7. These numbers are two times as high as the average number of deployed different classes and properties in the whole corpus.

Table 9.6: List of the 20 most common deployed schema.org properties of web sites using the class `s:Product` and/or `s:Offer`.

	Property	Web Sites #	Web Sites %
1	s:name	3 444	0.91
2	s:offers	2 767	0.73
3	s:description	2 744	0.72
4	s:price	2 744	0.72
5	s:image	2 737	0.72
6	s:url	1 937	0.51
7	s:availability	1 918	0.50
8	s:ratingValue	1 382	0.36
9	s:aggregateRating	1 378	0.36
10	s:priceCurrency	1 298	0.34
11	s:reviewCount	1 123	0.30
12	s:manufacturer	746	0.20
13	s:bestRating	677	0.18
14	s:productID	601	0.16
15	s:sku	552	0.15
16	s:brand	476	0.13
17	s:author	445	0.12
18	s:itemCondition	438	0.12
19	s:worstRating	426	0.11
20	s:reviewRating	358	0.09

Top Sites Table 9.7 lists the 20 most frequently visited web sites from this selection based on the ranking provided by Alexa. Although, except for `yr.no`, all sites belong to the *.com* top-level domain, the sites are from different geographic locations all over the world. `amarujala.com` is a Hindi news page, where `bdnews24.com` is an internet newspaper from Bangladesh and `thehill.com`is located in Washington DC. Also topical-wise those pages are diverse, where we find local and more global *classic*-newspaper topics (e.g., `palmbeachpost.com` and `upi.com`), but also more advisory-like sites (e.g., `rd.com`) and financial/business-related sites (e.g., `foxbusiness.com`). The geographical diversity indicates a variety of available news articles, without a bias towards a single region or political ideology.

Frequent Classes Similar as for the previous use case, we analyze in a first step the classes which are deployed by the use case-related sites. Table 9.8 presents the 20 most commonly deployed classes. In contrast to web sites related to a marketplace (compare Section 9.4.2), we cannot find one or two classes which are deployed by a larger fraction of all relevant web sites. In particular, only 28% of the web sites make use of the class `s:Article` which is meant to be used to annotate pieces of reports or news articles. Furthermore, around 36% of the site provide information about businesses and addresses, which for our news aggregator might in a first place not be helpful.

Frequently Co-occurring Classes Having detected the adoption of the different classes within the semantic annotations of news-related web sites, we want to identify semantic annotations of different types, provided by the same web site.

Table 9.7: List of the 20 most visited news-related sites making use of schema.org and Microdata within CC 2014 based on Alexa.

Rank	Web Site	Rank	Web Site
1	breitbart.com	11	bostonherald.com
2	yr.no	12	theroot.com
3	amarujala.com	13	mysanantonio.com
4	bdnews24.com	14	afr.com
5	thehill.com	15	parade.com
6	newsweek.com	16	weatherbug.com
7	upi.com	17	northjersey.com
8	mid-day.com	18	desmoinesregister.com
9	foxbusiness.com	19	palmbeachpost.com
10	rd.com	20	pressdemocrat.com

Table 9.8: List of the 20 most common deployed schema.org classes by news-related sites.

	Class	Web Sites #	Web Sites %
1	`s:LocalBusiness`	256	0.36
2	`s:PostalAddress`	255	0.36
3	`s:Place`	243	0.34
4	`s:WebPage`	241	0.34
5	`s:Article`	195	0.28
6	`s:AggregateRating`	194	0.28
7	`s:Store`	182	0.26
8	`s:Person`	182	0.26
9	`s:Thing`	135	0.19
10	`s:Product`	73	0.10
11	`s:Offer`	72	0.10
12	`s:Rating`	71	0.10
13	`s:Review`	71	0.10
14	`s:ImageObject`	70	0.10
15	`s:Event`	61	0.09
16	`s:VideoObject`	55	0.08
17	`s:Brand`	39	0.06
18	`s:Organization`	39	0.06
19	`s:Intangible`	34	0.05
20	`s:ImageGallery`	30	0.04

We therefore, similar to the former use case, calculate frequent itemsets. An excerpt of those itemsets is displayed in Table 9.9. Basically, we can identify three groups of sites. The first group of web sites embeds semantic annotations of the two classes `s:Article` and `s:Person`. Sites of this group mainly belong to the Australian media company *Faifax Media*, who operates those regional media sites. The second, making use of the class `s:Store` deploys in half of the cases also the class `s:LocalBusiness` and `s:PostalAddress`. Within this group, we cannot detect any obvious connections between the web sites. They belong to different media companies, are distributed all over the globe, and also do not use the same CMS. An example is `thestate.com` belonging to *Wanderful Media* using a plugin within their CMS which includes stores and products from `findnsave.com`. The products and stores are listed under the web site of `thestate.com` but actually belong to another web site. The third group of sites mainly makes use of the five classes, namely `s:LocalBusiness`, `s:PostalAddress`, `s:Place`, `s:WebPage`, and `s:AggregateRating` in any combination. We found that 24.3% make use of all the five classes together. Again, we inspected the content of the pages of those sites manually and tried to identify relations among them. A large fraction of those sites embedding the mentioned five classes is powered by the CMS *BLOX Content Management System* from `townnews.com`. This provider claims to be the "top CMS choice amount U.S. dailies"[106]. The current version of this CMS supports the annotation of news content using `s:Article` but in the corpus of 2014, these data was not annotated. Instead a plug-in, providing business listings and advertisements from other web sites, annotated the information using the mentioned five classes.

Article Information Although the class `s:Article` is not one of the most frequently used by semantic annotations from news-related web sites, it is the most relevant class for the use case of a news aggregator. We therefore inspect properties provided by semantic annotations of this class in more detail. We examine the most common used properties to identify which kind of information we can expect from semantic annotations of this class. Table 9.10 lists the 20 most commonly deployed properties for the mentioned class. The proportion again is related to the total number of sites making use of this class. We see that a large fraction (over 80%) of those sites annotate at least the name of the article, the content and the date the article was published. In 66% of the cases, we also find an author of the particular article. Besides, also 30% of the sites annotate an image for the article.

[106] `http://www.townnews365.com/news_room/study-shows-that-townnews-com-s-blox-cms-is-most/article_-d348b9a6-565f-11e5-9f65-b70fe7fc3b0c.html`

Table 9.9: Excerpt of the most interesting frequent itemsets of classes from news-related sites, ordered descending by their support.

Support	Itemset
.363	s:LocalBusiness
.362	s:PostalAddress
.345	s:Place
.342	s:WebPage
.326	s:LocalBusiness s:PostalAddress
.319	s:PostalAddress s:Place
.295	s:LocalBusiness s:Place
.295	s:LocalBusiness s:PostalAddress s:Place
.277	s:Article
.275	s:AggregateRating
.268	s:LocalBusiness s:PostalAddress s:AggregateRating
.267	s:LocalBusiness s:PostalAddress s:Place s:AggregateRating
.261	s:Place s:WebPage
.258	s:Store
.258	s:Person
.258	s:PostalAddress s:WebPage
.248	s:LocalBusiness s:WebPage
.248	s:LocalBusiness s:PostalAddress s:WebPage
.243	s:LocalBusiness s:PostalAddress s:Place s:WebPage s:AggregateRating
.206	s:Article s:Person
.165	s:LocalBusiness s:Store
.140	s:LocalBusiness s:PostalAddress s:Store

Table 9.10: List of the 20 most common deployed schema.org properties with the s:Article class by news-related sites.

		Web Sites	
	Property	#	%
1	s:name	184	0.94
2	s:datePublished	164	0.84
3	s:articleBody	160	0.82
4	s:author	129	0.66
5	s:image	58	0.30
6	s:copyrightHolder	33	0.17
7	s:description	21	0.11
8	s:url	17	0.09
9	s:interactionCount	6	0.03
10	s:headline	6	0.03
11	s:inLanguage	6	0.03
12	s:articleSection	5	0.03
13	s:dateCreated	5	0.03
14	s:keywords	5	0.03
15	s:video	4	0.02
16	s:dateModified	3	0.02
17	s:type	2	0.01
18	s:publisher	2	0.01
19	s:genre	2	0.01
20	s:aggregateRating	2	0.01

Table 9.11: List of the 20 most visited travel-related sites making use of schema.org with Microdata within CC 2014 based on Alexa.

Rank	Web Site	Rank	Web Site
1	hotels.com	11	royalcaribbean.com
2	agoda.com	12	iberia.com
3	aa.com	13	expedia.ca
4	priceline.com	14	cheaptickets.com
5	ryanair.com	15	choicehotels.com
6	orbitz.com	16	westjet.com
7	vrbo.com	17	onetravel.com
8	travelocity.com	18	airarabia.com
9	turkishairlines.com	19	hrs.de
10	hostelworld.com	20	venere.com

9.4.4 Travel Portal

Last, we want to study the utility of semantic annotations from the schema.org Microdata dataspace with respect to satisfy the information needs of a travel portal. Within the corpus, we found 299 web sites belonging to the category *Travel*, based on the list of web sites provided by Alexa. Those sites make on average use of 3 different classes per site, where the median is at 2, where over 34% of the sites only deploy one class. On average, the web sites from this category make use of 8 different properties (7.79), where the median is 5. In comparison to the average over all schema.org Microdata web sites the number of classes is slightly lower but the number of properties in almost two times as much. Meaning, that although only one type of entity is describe a larger set of properties is used, where we can expect richer descriptions.

Top Sites In Table 9.11, we list the 20 most frequently visited web sites in our corpus, which are categorizes as travel-related. Within this list, we find accommodation-booking sites like `hotels.com` or `hrs.com` as well as airline sites (e.g., `ryanair.com` and `westjet.com`) and also sites of full-service providers like `expedia.com` who offer bookings for flights, hotels, tours, and so on.

Frequent Classes Similar to the two previous studies, we first have a look at the classes which are deployed by the use case-related web sites. Table 9.12 shows the 20 most commonly used classes by those web sites. Similar to the news-related sites (compare Table 9.8), we cannot identify one or two dominant classes, which are embedded by all of those sites. Instead, we find around 30% of the sites annotating `s:PostalAddress`es and 18% annotating their information using the `s:Product` class. We also find 10% of the web sites providing semantic annotations of the class `s:Hotel`. Although we identified in the list of travel-related site the web sites of

Table 9.12: List of the 20 most common deployed schema.org classes by travel-related sites.

	Class	Web Sites #	Web Sites %
1	s:PostalAddress	83	0.28
2	s:WebPage	69	0.23
3	s:Thing	69	0.23
4	s:Organization	57	0.19
5	s:Product	54	0.18
6	s:AggregateRating	51	0.17
7	s:LocalBusiness	46	0.15
8	s:Article	43	0.14
9	s:Country	39	0.13
10	s:Hotel	29	0.10
11	s:Review	28	0.09
12	s:GeoCoordinates	28	0.09
13	s:BlogPosting	26	0.09
14	s:Person	25	0.08
15	s:Place	24	0.08
16	s:Rating	23	0.08
17	s:Blog	23	0.08
18	s:ImageObject	21	0.07
19	s:Event	20	0.07
20	s:Offer	18	0.06

aircraft carriers, we do not find the class `s:Flight` or `s:Airport` being deployed by any of those sites. A reason could be that the inclusion of those two classes and their related properties into the schema of schema.org aimed for the embedding of such information in reservations. Therefore, the implied usage of those classes might be a reason why we cannot find those information, as the crawler does not perform any bookings, so parts of the web sites might stay unexplored.

Frequently Co-occurring Classes Again, we calculate frequent itemsets, to detect which combinations of classes are embedded on travel-related web sites (compare Table 9.13). As already the most common classes are deployed by only around one quarter of all travel-related sites, we find the mostly supported itemsets of two or more items consisting of `s:PostalAddress` together with `s:LocalBusiness` or `s:Country`, embedded by 12%. 8% of the sites embed `s:PostalAddress` together with `s:Organization` or `s:AggregateRating`. 6% of the web sites providing information about hotels, also embed the class `s:PostalAddress`.

Hotel Information Going further, we analyzed the properties which are annotated in semantic annotations of class `s:Hotel`, as this class is most relevant for our use case. In Table 9.14, we list the 10 most common properties used together with the `s:Hotel`. Besides the name of the hotel, which is marked up by 72% of the sites, we find an address and an aggregated rating in 59% and 48% of the sites, respectively. Around 31% of the hotel-annotating sites also markup a description.

Table 9.13: List of the 20 most frequent itemsets of classes from travel-related web sites containing two or more items, ordered descending by their support.

Support	Itemset
.124	s:PostalAddress s:LocalBusiness
.124	s:PostalAddress s:Country
.080	s:PostalAddress s:Organization
.080	s:PostalAddress s:AggregateRating
.077	s:Thing s:Review
.077	s:Product s:AggregateRating
.074	s:Review s:Rating
.070	s:Thing s:AggregateRating
.067	s:Thing s:Organization
.067	s:Thing s:Rating
.067	s:Thing s:Review s:Rating
.064	s:PostalAddress s:Thing
.064	s:PostalAddress s:Hotel
.064	s:PostalAddress s:GeoCoordinates
.064	s:Thing s:Product
.064	s:AggregateRating s:Review
.064	s:LocalBusiness s:Country
.064	s:PostalAddress s:LocalBusiness s:Country
.060	s:PostalAddress s:Place
.060	s:AggregateRatings:GeoCoordinates

Table 9.14: List of the 10 most common deployed schema.org properties with the s:Hotel class by travel-related sites.

		Web Sites	
	Property	#	%
1	s:name	21	0.72
2	s:address	17	0.59
3	s:aggregateRating	14	0.48
4	s:geo	11	0.38
5	s:description	9	0.31
6	s:image	8	0.28
7	s:url	8	0.28
8	s:review	6	0.21
9	s:additionalType	3	0.10
10	s:priceRange	3	0.10

9.5 Discussion

In the following, we discuss the results of the former section with respect to the requirements we have identified for each of the three use cases (compare Section 9.1).

9.5.1 Marketplace

We can state that relevant semantic annotations (describing products and offers) are more frequently used by the use case-specific web sites than for the other two use cases. We found over 70% of the relevant web sites making exemplary use of the class s:Product. Over 70% of the web sites offering product and offer information also provide information about the name, description, the price as well as an image, which satisfy the information requirements identified for this particular use case.

As already mentioned before, only parts of the stated prices are accompanied by a specification of the currency. Although the value of `s:price` is defined as numeric, some web sites annotate the price including the currency, e.g., "12.45 EUR", as already found in a former study presented in [Meusel and Paulheim, 2015]. This issue needs to be further investigated for the final data fusion. Furthermore, we found over 50% stating the availability of the product, and 36% of the web sites annotate ratings. Product categories are not frequently provided. Of course, the data quality of property values needs to be investigated further, but based on the classes of deployed semantic annotations and the provided properties, the utility of the dataspace for the use case of a marketplace is sufficient.

9.5.2 News Aggregator

With respect to building a news aggregator. the results are also promising, although we found that only a small fraction of web sites (28%) providing news-related semantic annotations. Those web sites, providing semantic annotations of type `s:Article`, satisfy in 66% of the cases the identified information need of the use case. Even 30% of the web sites publishing articles using semantic annotations, provide a link to an image and 17% give copyright information. This is particular helpful when building the aggregator to avoid legal issues. Based on the findings of this empiric study, we can state, that semantic annotations from the schema.org Microdata dataspace fulfill the information requirements of a news aggregator.

9.5.3 Travel Portal

The results for the travel portal use case are disillusioning. Only 10% of the web sites provide annotated information about accommodations (in particular hotels). Information about flights or other services like car rental are not contained in semantic annotations.[107] The web sites providing semantic annotations for hotels, only in less than 50% of the cases annotate more information than the name and the address. Reviews are only provided by around 21% of the web sites, where only 10% state a price range for the hotel. Although some web sites provide all the required information, the total number of those is rather small. We therefore do not think that at this point the dataspace can satisfy the information need of the more complex use case of a travel portal.

Overall, we have shown that the utility of the dataspace depends on the use case and the functionalities which should be enabled by the use case. Consequently, a general utility analysis is not possible and for each particular use case a specific analysis is necessary. Exemplary for the news aggregator, we found a sufficient adoption to make use of articles from this dataspace, when we want to extend the functionality with a category-based filtering, the information are not covered by the dataspace.

[107] At least they are not annotated with the according classes based on the definition of schema.org.

Chapter 10

Product Data Categorization

In the previous chapter, the capability of semantic annotations to fulfill use case-specific information needs have been analyzed. It has been shown, that this capability depends on the use case, and in particular for the use case of a marketplace and a news aggregator the dataspace provides sufficient information. The presented analysis so far did only examine the semantic annotations and their utility for the use cases based on the deployed classes and properties, without further investigating in the values provided for the properties.

The examination of the values provided by semantic annotations for certain entities is an important step for the final integration and use of semantic annotations for a specific use case. Such an analysis cannot be performed globally, but only with respect to a concrete use case. In particular, in the following study, we want to identify the product categories of product-related semantic annotations contained in the dataspace. The motivation for this investigation is two-folded. First, insights into the distribution of product categories within the dataspace of product-related semantic annotations allows a fine-grained topical profiling of this specific part of the dataspace. Second, being able to (automatically) categorize product-related semantic annotations into a given product catalog enables the functionality of category-based search or filtering for the use case of a marketplace. Common state-of-the-art approaches, which (partly) solve the problem of classifying products into a given product categorization schema, require (large sets of) training data. The creation of such training data is in most cases resource intensive (money, workforce, time). Furthermore, with respect to a fine-grained topical profiling, for each topical domain a dedicated set of training data would be required. In order to bypass the costly process of creating training data, we analyze to which extend web site-specific product category information can be exploited to assign product-related annotations to the correct position within a given product classification schema.

In the following the problem is described in more detail, where the subsequent section summarizes related work in the area of product classification. Section 10.3 explains the data which is used in order to carry out the study and the method which is used to evaluate the results. In a consecutive way, the applied methods and the

results are presented in Section 10.4. The chapter concludes the results in the last section and compares them to supervised approaches, which are state-of-the-art.

The approach and the evaluation presented in this chapter have already been published in [Meusel et al., 2015b].

10.1 Problem Description

The overall goal of this chapter is to assign product-related semantic annotations into a given product categorization schema. This is useful for a more fine-grained topical profiling of this part of the dataspace. Furthermore, an approach which is capable of assigning for a new product-related semantic annotations the corresponding product category, can be used by a marketplace to enable the functionality of category-based filtering and search.

Unfortunately, the definition of schema.org does not provide any further subclasses of the class `s:Product`. Therefore, we cannot simply map those subclasses towards a given product classification schema.

Inspecting the product-related semantic annotations, we found that some of them make use of the properties `s:category` and `s:breadcrumb`. Where the first one is directly design to allow web sites to include their own, web site-specific categorization tree for a product. The later allows to annotate the navigation tree to the particular page. As shopping sites mostly are organized based on their product categories, the navigation tree is similar to the product categorization. Unfortunately, as said before, these categorizations are web site-specific. Therefore, they do not follow a unique classification schema. Furthermore, only a small fraction of web sites embeds those properties.

State-of-the-art approaches mostly exploit existing product catalogs to train supervised classification models. In some cases, those product catalogs are created internally (as they support an existing application, e.g., shopping site) or they are bought from a third-party provider. In our case, we neither own such a corpora, nor are willing to pay for a training dataset.

Therefore, in the following we explore to which extend we can make use of the categorization information provided by the semantic annotations from the different web sites, in order to assign a product category. We use a distant-supervised approach, which does not require a training corpora.

10.2 Related Work

The automatic classification of products, based on a given set of target categories, has been extensively studied since the rise of e-commerce platforms and available product offers in the Web. In [Ding et al., 2002], the authors describe the system named *GoldenBullet* which is one of the first end-to-end solutions in order to pre-process data, learn a classification model from pre-labeled data and apply this

model on unseen product offers. The authors report a maximal accuracy for non-hierarchical classification using a *NaiveBayes* classifier of 78%. Another, more recent approach is presented in [Petrovski et al., 2014], where the authors made use of product attributes and their categories retrieved from `amazon.com`, in order to train a classification model for unseen, *non-amazon.com* products. This approach could also be used for our case, but it is questionable to which extend it is legal to use the products and categorization provided by amazon for non-commercial use cases. A similar approach is also proposed by [Köpcke et al., 2012]. They use different pre-learned feature extraction methods and similarity methods to match equal products. They reach an overall F1-measure of 0.66, evaluated on over $100k$ product offers. Unfortunately, they do not report the e-commerce portal(s) used for training and hence a direct comparison is not really possible. In addition, the data is not available to the public.

An approach, making use of features retrieved from images, is presented by [Kannan et al., 2011b], where they show how the relatively weak signals from product images can be leverage to improve precision of product classification by up to 12%.

A recent approach by [Qiu et al., 2015] presents a system which efficiently detects product specifications from product detail pages for a given product category. In order to determine the category, they make use of pre-trained classification models and a set of seed product identifiers of products related to this category.

[Bakalov et al., 2011] focus also on the extraction of attribute values for a given set of entities from a specific category. In particular, they focus on numerical and discrete attributes. They make use of overlapping entities from different documents in order to extract new key value pairs using an *Integer Linear Program* (ILP). Their overall goal is to expand the product catalog and by this the commerce search engine using the Bing Shopping data.

[Nguyen et al., 2011] present an end-to-end solution facing the problem of keeping an existing product-catalog with several categories and sub-categories up-to-date. Their approach includes data extraction, schema reconciliation, and data fusion. The authors show that they can automatically generate a large set of product specifications and identify new products.

In the work of [Kannan et al., 2011a], the authors focus on the matching of unstructured product offers from potentially thousands of data providers to a given structured product catalog. Their approach learns a matching function off-line which is later applied during runtime to match newly discovered offers into the given target taxonomy. In particular, they parse offer descriptions and extract several known feature value pairs, which are used to train a matching model. In their approach, they underline the importance of the differentiation between a mismatch and a missing value. As stated their approach makes use of a populated product catalog, where they also use the Bing Shopping dataset.

An interesting approach is presented in [Papadimitriou et al., 2012]. The authors try to tackle the problem of integrating products from different providers (like *Amazon* and *pricegrabber*) into one target taxonomy. The reformulate their

problem also into an ILP and make use of the taxonomy of each provider. Their approach is highly comparable to ontology matching approaches like the one presented in [Udrea et al., 2007]. They also assume, as the approaches mentioned upfront that the target ontology is already populated with data and the data is more or less richly described. Their experiments are based on three providers and one target ontology, where they restrict themselves to *consumer electronics*, which they manual filter upfront. In their runtime experiments, they report a linear dependency between the number of products in the ILP and the number of datasets and categories. This linear dependency is a good indicator in order to also try to solve the problem of product classification without a populated target taxonomy. In our approach, we also make use of ILP, but as the later results show, the claim of [Papadimitriou et al., 2012] that the approach applicable to web-scale, is questionable.

All the mentioned approaches make use of hand-labeled or pre-annotated data. As such data is not (easily) accessible in larger quantities, and always goes together with the dependency on an external data classification provider, in the following proposed approach, we try to overcome this necessity and propose an alternative method to create labeled training data. Furthermore, most of the data which was used for the named approaches, is not available to the public, which makes a comparison difficult.

10.3 Research Data and Evaluation Method

In this section, we first describe the subset of the semantic annotations which is used as input data for our approach. We further briefly discuss the target product classification schema, as well as the processes employed to create a goldstandard dataset for evaluation. The last part of this section describes the baseline as well as the evaluation method.

10.3.1 Product schema.org Microdata

From the cleaned data corpus, generated by the approach described in Chapter 7, we derived a subset of 9 414 product entities, originating from 818 different data providers. Those product entities are annotated at least with the properties `s:name`, `s:description`, and `s:brand`. In addition, either one of the two properties `s:category` (84% of the semantic annotations) or `s:breadcrumb` (16% of the semantic annotations) is annotated. From each data provider, we extracted at most 20 product entities to overcome the potential bias towards a certain category. Table 10.1 shows an excerpt of the data. Especially for the categories/breadcrumb values, we observed a mixture of multi-level and flat paths, as well as tag-like annotations. 3 653 distinct `s:category` values and 1 019 distinct `s:breadcrumb` values are used by the included products.

Table 10.1: Examples of product-related semantic annotations from Microdata with schema.org making use of the `s:category` and/or `s:breadcrumb` property.

`s:name`	`s:decription`	`s:brand`	`s:category/ s:breadcrumb`
ColorBox Stamp Mini Tattoo	ColorBox Stamps are easy to use and perfect for papercraft fun. [..] Not for use by children 12 years and younger.	ColorBox	Stamps >Rubber Stamp
Cowhide Link Belt	ITEM: 9108 Your search is over for a great casual belt for jeans or khakis. [..]	-	Accessories
Fiesta SE	Automatic, Sedan, I4 1.60L , Gas, RedVIN: 3FADP4BJ8DM1679	Ford	cars
Alabama Crimson Tide Blackout Pullover Hoodie - Black	No amount of chilly weather can keep you from supporting your team.[..]	-	Alabama Crimson Tide $>40to60$
231117-B21 HP PIII P1266 1.26GHz ML330 G2	Description:Pentium III P1266 ML330 G2/ML350 G2/ML370G2 (1.26GHz/ 133MHz/512K/43W) [..] # 231117-B21	HP Compaq	G2 Xeon
TFS Lil' Giant Anvil, 65 lb	Dimensions: Face 4" x 10.75" Horn 4" x 8.25" Height8" Base 9.25" x 11" Hardie Hole: 1" [..] #: TFS7LG65	Anvils [..]	Hardware >Tools >Anvils
Gavin Road Cycling Shoe	For great performance at a discounted price, [..]	-	Root RoadBikeOutlet.com >Apparel >Shoes >>

10.3.2 GS1 - Global Product Catalogue

For our experiments, we used the *GS1 Product Catalogue* (GPC) as target product hierarchy. The GPC is available in different languages and claims to be a standard for everything related to products.[108] The hierarchy is structured in six different levels starting from the *Segment*, over *Family*, *Class*, and *Brick*, down to the last two levels *Core Attribute Type* and *Core Attribute Value*. The first level distinguishes between 38 different categories, the second level divides the hierarchy into further 113 categories and the third level consists of 783 disjunct categories. In addition to the textual labels for each category in the hierarchy, the forth and the sixth level partly include a – more or less – comprehensive textual description. Table 10.2 shows the first four levels of three selected paths of the hierarchy.

Table 10.2: Excerpt of GS1 GPC (first four levels). [..] is a placeholder, if the label is similar to the one of the former level.

Segment	Family	Class	Brick
Toys/Games	[..]	Board Games/Cards/Puzzles	Board Games (Non Powered)
Food/Beverage/Tobacco	Seafood	Fish Prepared/Processed	[..] (Perishable)
Footwear	[..]	Footwear Accessories	Shoe Cleaning

[108] `http://www.gs1.org/gpc`

An interesting development is the publication of one of the first external schema.org extension by GS1 to allow a more fine-grained annotation of product-related information [Brickley, 2016]. GS1 provides through their new extension their product categorization schema as additional subclasses, e.g., `gs1:Beverage` or `gs1:Clothing` of `s:Product`. As so far data providers only annotated their products using `s:Product`, an automatic or semi-automatic more fine-grained annotation of the existing products would potentially increase the spread of this external vocabulary and improves the possibility for more detailed profiling of the product-related dataspace.

10.3.3 Product Goldstandard

Using the set of categories from the previously mentioned hierarchy, we manually annotated the set of products described in Section 10.3.1. Specifically, we label each product (if possible) with one category for each of the first three levels. The annotations were performed by two independent individuals. Whenever there was a conflict, a third person was asked to solve the discrepancy. The annotators were first asked to read the title/name, description, and the additional available values of the product and, in case of insufficient information, they should visit the web page of the product.

In the goldstandard, we could not assign any category to 187 (2.09%) products, mostly because the available attributes to describe the products were insufficient, and the web page of the product was either not available any more or here also not enough information were given. Based on the first level of the GS1 GPC hierarchy, we assigned at least each category once (except *Cross Segment*). Table 10.3 depicts the ten most frequent categories of the first level within the goldstandard. We see a dominance of the category *Clothing*. For the second level, we assigned 77 (68.14%) different labels at least once, and 303 (38.70%) different labels for the third level. The goldstandard, as well as more comprehensive statistics, can be found as part of the WDC project.[109]

10.3.4 Baseline and Evaluation

As we want to examine to which extent the categorizations of the single web sites can be used to assign categories from a global hierarchy to products, we compare our results to the results of a supervised classification approach. The approach is trained using 10-fold cross-validation with three different classification approaches: Naive Bayes (NB), Decision Trees (DT) and k-Nearest Neighbor approach, where $k = 5$ (5-NN). A detailed description of the baseline method can be found in Section 10.4.2.

For reasons of comparison, we use *accuracy* ($\mathcal{ACC}$) as the main evaluation metric. Whenever an approach is not able to return a category for a given product,

[109] `http://webdatacommons.org/structureddata/2014-12/products/gs.html`

Table 10.3: Distribution of categories for the first level of the GS1 GPS within the goldstandard dataset, accompanied by the predicted category distributions of the best supervised and distant-supervised approach.

Rank	Category Level 1	Original	Superv.	Δ	Dist. Superv.	Δ
1	Clothing	.435	.401	.033	.406	.028
2	Personal Accessories	.053	.128	.075	.039	.014
3	Household/Office Furniture/Furnishings	.051	.045	.006	.035	.016
4	Automotive	.047	.054	.007	.052	.005
5	Computing	.037	.034	.004	.023	.014
6	Audio Visual/Photography	.036	.030	.006	.020	.015
7	Healthcare	.033	.027	.006	.005	.027
8	Pet Care/Food	.026	.028	.002	.017	.010
9	Sports Equipment	.026	.030	.004	.022	.004
10	Food/Beverage/Tobacco	.024	.025	.001	.007	.018
11-38	*Others*	.232	.198	.065	.373	.159

we count this examples as a *false negative*. For approaches returning either one label or no label for each instance, this measure is equal to recall ($\mathcal{R}$). In addition, for our distant-supervised approaches, we also report the precision ($\mathcal{P}$), as this measure gives an idea about the performance of predicted labels, without regard of those which cannot be labeled. We also state the F-score $\mathcal{F}1$, representing a trade-off between $\mathcal{R}$ and $\mathcal{P}$.

10.4 Distant-Supervised Product Categorization

In this section, we first state how both input data sources, i.e., product descriptions and categories from the given target hierarchy, are transformed into feature vectors, that can be processed by further methods. Then, we train a model based on the hand-annotated categories of the goldstandard. In the remaining part, we describe the consecutive improvements of our distant-supervised approach, making use of the categorical information for the products given in the semantic annotations.

10.4.1 Feature Vector Generation

As stated above, we have two types of input: products, which are described by a set of properties, and the categories of the target hierarchy. In order to transform both types of input into comparable feature vectors, we generate a *bag of words* representation for each entity, i.e., each product and each category at a certain depth within the hierarchy.

For the products, we experiment with different sets of property combinations (e.g., only `s:title`, `s:title` with `s:description`, and so on). For the hierarchies, we use the textual names of the categories themselves and all or a selection of the names of sub-categories (e.g., segment, segment and family, segment and brick). In all cases, we tokenize the textual values by non alpha-numeric characters, remove stopwords and stem the tokens using a *Porter Stemmer* [Porter, 1980]. Moreover,

we transform all characters to lower case and remove terms which are shorter than 3 and longer than 25 characters.

In order to weight the different features for each of the elements in the two input sets, we apply two different strategies:

Binary Term Occurrence (BTO), where the weight of a term for an element is either 1, if the term occurs at least once within the textual attributes of the element, 0 otherwise.

TF-IDF, where the term frequency is normalized by the inverse document frequency, which removes the impact of common terms which occur in a large fraction of documents.

In the following, we refer to the set of feature vectors describing products by Pro and to those describing labels of the categories of the hierarchy by Cat. Depending on the textual attributes which were used to create the vectors, the number of final attributes ranges between 4 000 (only category and breadcrumb) to around 11 000 (all properties).

10.4.2 Baseline: Supervised Approach

Table 10.4 presents the results with different setups for the baseline classification approach. We reach the highest accuracy with a 5-NN classification algorithm using *Jaccard Coefficent*. Decision Trees do not perform at a comparable level, so we exclude them from the table. We also calculate the distribution of the predicted product categories for the best approach. The results are shown in column four and five in Table 10.3. Column four states the percentage of entities classified according to the corresponding category. Column five measures the distance of the percentage to the original percentage of entities of the corresponding category in the goldstandard.

10.4.3 Hierarchy-Based Product Classification

In a first step, we use the feature vectors created for the categories from the target hierarchy Cat in order to train a predictive model (one labeled example for each category). This model is then used to predict the labels for the instances of Pro. We test different classification methods, namely *Naive Bayes* (NB), *k-Nearest-Neighbor* with $k = 1$ (1−NN)[110], *Support Vector Machines* (SVM), and *Random Forests* (RF).

[110] As for each class, only one example exists k needs to be set to 1, otherwise the method would consider other examples then the nearest, which by design belong to another class. This setup is equal to *Nearest Centroid Classification*, where each feature vector of Cat is equal to one centroid.

Table 10.4: Selected results of the baseline classification for assigning GS1 GPC first level categories. Highest score is marked in **bold**.

Selected Properties	Term Weighting	Classifier	$\mathcal{ACC}$
Name,Description	BTO	NB	.722
Name,Description	TF-IDF	NB	.733
Name,Description	BTO	5-NN(Jaccard)	.608
Name,Description	TF-IDF	5-NN(Cosine)	.728
Name,Description	BTO	DT	.366
Name,Description	TF-IDF	DT	.363
Name,Description,Category,Breadcr.	BTO	NB	.754
Name,Description,Category,Breadcr.	TF-IDF	NB	.757
Name,Description,Category,Breadcr.	BTO	5-NN(Jaccard)	.819
Name,Description,Category,Breadcr.	TF-IDF	5-NN(Cosine)	.740
Name,Description,Category,Breadcr.	BTO	DT	.367
Name,Description,Category,Breadcr.	TF-IDF	DT	.363
Name,Description,Category,Breadcr.,Brand	BTO	NB	.758
Name,Description,Category,Breadcr.,Brand	TF-IDF	NB	.760
Name,Description,Category,Breadcr.,Brand	BTO	5-NN(Jaccard)	**.820**
Name,Description,Category,Breadcr.,Brand	TF-IDF	5-NN(Cosine)	.746
Name,Description,Category,Breadcr.,Brand	BTO	DT	.367
Name,Description,Category,Breadcr.,Brand	TF-IDF	DT	.363

Table 10.5 shows the results of the best configuration, using only the features from the values of the properties name, category and breadcrumb from Pro and all hierarchy labels from the GS1 GPC. We find that on average TF-IDF as term weighting methods performs better than a BTO strategy. The best results are achieved using 1-NN and Naive Bayes classification on TF-IDF vectors.

10.4.4 Similarity-based Product Category Matching

In order to exploit the promising performance of the distance-based classification approach (1-NN) of the former section, we extend our approach in this direction, using the similar fundamental idea as Nearest-Neighbor classifier. We calculate for each instance in Pro the distance to all instances in Cat. To that end, we use three different similarity functions, namely:

Cosine Similarity: This measure sums up the product of the weights/values for each attribute of the two vectors and is supposed to work well with TF-IDF.

Jaccard Coefficient: This measure calculates the overlap of terms occurring in both vectors and normalize it by the union of terms occurring in both vectors. This measure is supposed to work well with binary weights.

Non-normalized Jaccard Coefficient: As the description of products can be rather comprehensive (based on the way the data is annotated), we address the penalization of longer product names, which would occur for Jaccard, by introducing a non-normalized version of the *Jaccard-Coefficent*, i.e., only measuring the overlap of tokens.

In addition, we use different sets of textual attributes from the products as well as from the hierarchies to create the feature vectors. Based on the similarity matrix,

Table 10.5: Best results achieved with distant supervised classification using instances of Cat for training. Highest scores are marked in **bold**.

Term Weighting	Classifier	$\mathcal{ACC}$
TF-IDF	NB	**.377**
TF-IDF	1-NN (Cosine)	**.377**
TF-IDF	1-NN (Jaccard)	.361
TF-IDF	SVM	.376
TF-IDF	DT	.025
TF-IDF	RF	.006
BTO	NB	.000
BTO	1-NN (Cosine)	.330
BTO	1-NN (Jaccard)	.271
BTO	SVM	.000
BTO	DT	.025
BTO	RF	.026

we then select for each instance in Pro the instance in Cat with the highest score, larger than 0. In contrast to a classifier, we do not assume any distribution within the data, or assign any category randomly. Meaning, in case of two or more possible categories which could be assigned, we do not assign a particular instance from Cat to the instance of Pro.[111]

Table 10.6 presents a selection of results of this approach, trying to predict the categories of the first, second and third level within the hierarchy. In each of the three blocks, the first line always reports the best results using only the category and breadcrumb as input for the feature vector. The second line reports the result for the default configuration (all attributes, TF-IDF). The third line shows the result for the optimized setup of attributes and term weighting. Overall, cosine similarity performs best. For some configurations, the other two tested similarity functions produce comparable results, but overall do perform worse than cosine similarity. Starting from level one to three, we see a slight decrease in terms of accuracy. This is not surprising as the number of possible labels increases with each level (see Section 10.3.3) and the contentual boundaries between them become more and more fuzzy. In addition, we find that the best configuration just differs by some percentage points from the default configuration for all three levels (e.g., .341 vs. .359 for the first level). Furthermore, using only the information from the category and the breadcrumb alone does not produce the highest accuracy results. For all three levels the best results in terms of accuracy can be reached using the textual values of category, breadcrumb and name as input for the feature vector creation.

Inspecting the results of the optimal solution for each level manually, we find that in most cases the overlap in features between the instances of Pro and Cat is insufficient for those instances which were wrongly categorized or left unlabeled. Reasons for this are the use of a different language as the target hierarchy (e.g.,

[111] As stated before, such instances are counted as *false negatives* within the evaluation.

Table 10.6: Selected results for all three category levels, including the default configuration, the best with and without ground knowledge.

Product Properties	Weight.	Hierarchy Levels	Term Weight.	Ground Knowldg.	$\mathcal{ACC}$	$\mathcal{P}$	$\mathcal{F}1$
Level 1							
Category, Breadcr.	BTO	1-6	TF-IDF	none	.288	.334	.309
All	TF-IDF	1-6	TF-IDF	none	.341	.344	.343
Name, Category, Breadcr.	BTO	1-6	TF-IDF	none	.359	.373	.366
Name, Category, Breadcr.	BTO	1-4	TF-IDF	DISCO	.479	.499	.489
Level 2							
Category, Breadcr.	BTO	1-6	TF-IDF	none	.171	.297	.217
All	TF-IDF	1-6	TF-IDF	none	.261	.264	.263
Name, Category, Breadcr.	BTO	1-6	TF-IDF	none	.294	.305	.300
Name, Category, Breadcr.	BTO	1-4	TF-IDF	DISCO	.380	.395	.387
Level 3							
Category, Breadcr.	BTO	1-6	TF-IDF	none	.109	.112	.111
All	TF-IDF	1-6	TF-IDF	none	.196	.198	.197
Name, Category, Breadcr.	BTO	1-6	TF-IDF	none	.257	.267	.262
Name, Category, Breadcr.	BTO	1-4	TF-IDF	DISCO	.258	.269	.263

Spanish), a different granularity (e.g., *fruits* versus *cherry*) or the use of synonyms (e.g., *hat* versus *cap*).

A common method to overcome at least the two latter discrepancies is the enhancement with external/additional ground knowledge. For our experiments, we use two different sources of ground knowledge to enhance our feature vectors. First, we make use of the *Google Product Catalog*.[112] This catalog is used by Google to enable advertisers to easily categorize the ads they want to place. The catalog is available in different languages and in addition, is more tailored towards products traded in the Web. The second source we use is based on the co-occurrences of different terms within a corpus. In particular, we make use of *extracting DIStributionally related words using CO-occurrences* framework (DISCO)[113], first presented by [Kolb, 2008], where we have used the English Wikipedia to enhance the feature vectors of the categories.

The best results and the comparison to the best results without the enhancement can also be seen in Table 10.6, within the third and forth row of each block. In general, we find a strong increase in the accuracy in comparison to the non-enhanced experiments. For the first level, we increase our performance by 33% to almost 50% accuracy. For level three, however, this effect diminishes almost completely. Even with the enhanced vectors, the improvements are small.

In the following, we describe two different types of experiments to further improve our results. In the first, we concentrate on high-precision results and obtained those values as labeled instances. Then, we train a predictive model on those instances. In the second approach, we reformulate the task of labeling a set of instances as a global optimization problem, following the idea presented in [Bakalov et al., 2011].

[112] `https://support.google.com/merchants/answer/1705911?hl=en`
[113] `http://www.linguatools.de/disco/disco_en.html`

10.4.5 Classification on High-Precision Mappings

This approach is based on the idea that, even if the accuracy (which represents the global performance of the matching) is not sufficient, we could make use of those instances which were assigned to a category with a high precision. Those instances can further be used as input in order to train a predictive model. It is important to note that when selecting the mapping, all, or at least a large fraction of categories (which should be predicted), should be included. This means that some configurations even with $\mathcal{P} = 1$ are not useful, as they include too few instances. In order to improve the precision of our initial mapping, we introduce a higher minimal similarity between products and categories.

The first columns in Table 10.7 show the highest precisions which could be reached, where at least 100 product instances were assigned to an instance of Cat of level 1. The precision of those optimal configurations ranges between $.75$ and $.79$, which means that within this data, every fourth or fifth instance is wrongly labeled. In addition, we report the values for a less precise setup ($.57$) but with over $5\,500$ labeled examples. We tested different mappings and train different classification methods on this input data.

In Table 10.7 we outline the best performing results for the different input configurations.[114] Compared to the results based on the classification model learned on the data depicted from the hierarchy (compare Table 10.5), we observe improvements up to an overall accuracy of 51%. But still the results are 30% worse than a supervised approach.

Table 10.7: Result of combined approach, using high-precision mapping result as training data to learn a predictive model for first level categorization.

Configuration: Product Properties (Weight.) | Hierarchy Level (Weight.) | Ground Knowldg.	Min. Sim.	Mapping $\mathcal{ACC}$	Mapping $\mathcal{P}$	Mapping # Inst.	Overall Classif.	Overall $\mathcal{ACC}$
Name,Cat.,Breadcr. (BTO)|1-6 (TF-IDF)|Google	>.35	.009	.789	109	NB	.076
All (BTO)|1-6 (TF-IDF)|Google	>.25	.008	.772	103	5-NN	.079
Name,Cat.,Breadcr. (TF-IDF)|1-6 (TF-IDF)|Google	>.25	.028	.747	340	NB	.069
Name,Cat.,Breadcr. (BTO)|1-4 (TF-IDF)|Disco	>.05	.340	.570	5 505	DT	.514

We find, that in case of the high-precision configurations (first three rows) the overall precision of the classifier which can be trained based on those input data is poor, and in all three cases did not exceed 10% overall accuracy. Manually inspecting those datasets and the resulting classifications reveals that not all classes are contained in those sets. Consequently, the model cannot predict all classes (as they are unknown) and that the number of training data is not enough even for the classes which are included. Inspecting the results of the fourth configuration, where the final accuracy exceeds slightly the 50%, we find almost a balanced distribution in the errors of the classification.

[114] We also applied up-sampling of under-represented classes in the dataset, but the results did not improve.

10.4.6 Global Optimization

In the presented approaches so far, we evaluate each match between an instance in Pro and Cat in isolation. However, the similarity between two products should be used as an indicator for mapping these instances to the same category, and vice versa. Deciding about the similarity of products and matching them to categories are thus highly dependent problems.

We try to take these dependencies into account by formalizing the problem as a global optimization problem in terms of Markov Logic [Domingos and Lowd, 2009]. In particular, we use the solver *RockIt* by [Noessner et al., 2013] to compute the most probable solution, also known as the MAP (maximum a posteriori) state. In our formalization, we use two weighted predicates *map* and *sim* to model the mapping of a product to a category and to model the similarity of two products. We use the similarity matrices from the former experiments as input for defining the prior weights for the respective atoms. Then, we define the score which needs to be optimized as the sum the weights attached to these atoms. Further, we define two hard constraints which have to be respected by any valid solution. (1) If two products are similar, they have to mapped to the same category. (2) Each product can be assigned to only one category.

We used the best configuration of the former similarity-based matching results from Section 10.4.4, where we reached an accuracy level of $.479$. We tested different combinations for the similarity of products, as well as the minimal similarity, we employed into the global optimization problem. In addition, we also tested different weight ratios between the two predicates, where we multiply the original weight of *map* with a constant factor. In Table 10.8, we report the best configurations and the corresponding accuracy values. In comparison to the original value of $.479$, we can improve up to $.555$, and we assume that this is not the best value which can be reached. Unfortunately, even when running the solver on a large machine (24 cores and over 300GB RAM), further experiments cannot be finished within 24 hours, which shows that the approach is promising, but requires more tweaks to run at large scale.

We selected the best performing distant-supervised approach and calculated again the resulting distribution of product categories of the products contained in the goldstandard (see column 6 and 7 of Table 10.3). Note that the supervised approach has a summed absolute error of $.20$ while the best distant supervised approach has an error of $.31$ (the average absolute error is $.006$ and $.008$, respectively).

10.5 Summary

In this chapter, we have tried to gain further, more fine-grained topical insights into product-related semantic annotations. We manually annotated a subset of this data with categories of the first three levels of the GS1 Global Product Catalog. Based on that goldstandard, we have shown that supervised methods can reach an accuracy of 80% when learning a predictive model in order to categorize products.

Table 10.8: Results of the best configurations for solving the optimization problem. Highest scores are marked in **bold**.

similarity map	similarity sim	min. value for sim	weight ratio map/sim	$\mathcal{ACC}$	$\mathcal{P}$	$\mathcal{F}1$
Cosine	Cosine	0.5	20/1	.505	.540	.522
Cosine	Jaccard	0.5	20/1	.483	.506	.494
Cosine	Cosine	0.5	10/1	.514	.556	.534
Cosine	Jaccard	0.5	10/1	.484	.509	.496
Cosine	Cosine	0.4	10/1	.553	.606	.578
Cosine	Cosine	0.3	10/1	**.555**	**.636**	**.593**

Further, as already some sites mark products with web sites-specific category information, we first have shown that using this information alone, due to its heterogeneity among different sites, is not an optimal input for a distant-supervised approach. But in combination with other properties (e.g., the name), that information can be leveraged by distant-supervised methods and thereby assign categories to a given set to products with an accuracy of up to 56%. To that end, we use various refinements of the problem, taking both background knowledge into account, as well as modeling the categorization of a set of instances as a global optimization problem. The latter provides very promising results, but also hints at scalability issues of solving such optimization problems. This is contrary to the observations of [Papadimitriou et al., 2012], who claim to be able to solve the global matching task on an ordinary server over $500k$ products within less than one hour and report a linear dependency between the number of products and the necessary time. A promising work was presented by [Gonina et al., 2011] in terms of scalability using *MapReduce*-based distributed computation engines, which might be able to overcome the scalability issue.

Regarding the distribution of product categories which are predicted by the two different kinds of approaches, we see that the supervision works slightly better, but both results can be used in order to gain first insights in the category distribution of the dataset. Here, it is important to keep in mind that the goldstandard, due to its size, as well as the training data might not be representative for the whole Web and therefor the performance of the approaches might be different for semantic annotations from other parts of the Web.

Another area where further improvements can be made is the selection of sources. In our goldstandard, we only included product descriptions from less than $1\,000$ different web sites, while in the Web, there are by far more which can be exploited. In particular, it might be a promising approach to weight the influence of products of a particular web site by other attributes, for example the average length of the description or the depth of the given category information.

Chapter 11

Conclusion

This chapter summarizes the three previous parts of the thesis, outlines the major contributions, and values them with respect to the goal of the thesis. Subsequent, we discuss the research impact of the major contributions presented in this thesis.

The first two chapters have introduced the reader to the dataspace of semantic annotations embedded in HTML pages using Microformats, RDFa, and Microdata. We have outlined potential use cases where semantic annotations can be a valuable data input source. But, we also emphasized that until the writing of this thesis, only little (public) work has been done in the field of profiling this particular dataspace, which enables a further estimation of its potential for concrete use cases or applications. In order to overcome the lack of knowledge about semantic annotations, the thesis has addressed the problem in three consecutive steps:

11.1 PART I: Extraction of Semantic Annotations

In the three chapters of the first part of the thesis, we have focused on possibilities to gather, in an efficient way, semantic annotations from the Web. In particular, two different approaches have been discussed.

First, it has been shown, how focused crawling can be used in order to steer a crawler directly to find web pages embedding semantic annotations in general. Furthermore, the adaptability of the strategy for more fine-grained objectives, e.g., the collection of semantic annotations from the Web containing richer descriptions, has been demonstrated. The state-of-the-art focused crawling strategies have been extended by a bandit-base selection strategy resulting in a refinement learning approach. This approach enables the crawler during the process of collecting web pages to find the pages, which increase the overall fitness. We have shown that the approach outperforms a breadth-first search strategy by factor two and in comparison to a pure online-based strategy by 26% in terms of the achieved harvesting rate, as presented in Section 3.4. The results demonstrate, that focused crawling can be used to efficiently collect (information rich) semantic annotations from the Web.

Furthermore, it especially allows the efficient discovery of web sites, providing semantic annotations.

Second, the thesis has introduced a framework, which enables the collection of semantic annotations directly from existing web corpora, such as the corpora provided by the *Common Crawl Foundation*. The framework is vertical and horizontal scalable and can be execute within a cloud-computing environment. Using the framework, semantic annotations have been extracted out of three different web corpora containing over 7 billion HTML pages (Section 4.3). The scalable fashion of the framework together with the ability to be executed in an on-demand environment enables also non-profit organizations like universities to parse and extract information from terabyte-scale data sources.

Within the final chapter of this part, we have focused on the aspect of data representativity. Although this specific aspect is crucial in order to draw reliable conclusion from samples, it is not often addressed in scientific publications working with data from web corpora. For the two presented data extraction approaches, we have comprehensively discussed the representativity of the thereby collected data for the whole Web. With respect to the goal of the thesis, we have shown that the data extracted by the second approach is more feasible in order to allow statements about the overall spread of semantic annotations in the Web. In addition, we have shown that the expected maximum sampling error, based on the size and fraction of the sample, is sufficiently small enough to allow reliable statements (compare Section 5.2).

Nevertheless, both approaches are capable to discover and extract semantic annotations from the Web. Based on the subsequent use case, the one or the other might be preferable. Although both approaches allow the discovery of web site embedding semantic annotations, both are barely capable of collecting all semantic annotations embedded by a single web site. For such use cases, other approaches, restricting the crawler to a set of selected web sites, might be more applicable.

11.2 PART II: Analysis of the Deployment of Semantic Annotations

The chapters of this part of the thesis have focused on the use case-independent profiling of the adoption of semantic annotations in the Web. Furthermore, the topical coverage of semantic annotations as well as its data quality and evolution over time have been discussed.

As one of the first, the thesis has presented recent statistics about the adoption of semantic annotations in the Web, embedded by the three semantic markup languages Microformats, RDFa, and Microdata. The empirical findings show that in the last years, from 2012 to 2014, the percentage of web sites making use of semantic annotations has strongly increased from 5.6% in 2012 to 25% in 2014 (Section 6.2). Furthermore, the topical profiling revealed that especially product-, location-, organization-, and website-related information are described by semantic

annotations. Due to the increasing spread of semantic annotations in the Web and the broad topical coverage, we think that semantic annotations are potentially interesting for a large set of different applications and use cases.

Within the first analysis of this dataspace, we identified two issues which influence the result of the profiling. The first are duplicated entities, which results for example from headers or footers in web pages of the same web site. The second are violations of the definition of the used vocabulary. The thesis has presented a set of high precision heuristics to overcome a large fraction of such violations. Those heuristics, together with a RDF-specific duplicate detection are implemented in a pipeline within this thesis. We have shown that these rather simple, but efficient data cleansing steps have a drastic effect on the retrieved profile of the Microdata schema.org dataspace. In particular, as we have shown in Section 7.3, the overall number of described entities is reduced by 60%. The findings underline that also in the dataspace of semantic annotations quality issues exist, but that they can be compensated by straightforward, dataspace-specific approaches.

Another investigation presented in this part of the thesis has focused on the evolution of the adoption of semantic annotations embedded in HTML pages using Microdata together with schema.org, in more detail. As one of the first, at the point of writing, we have analyzed the interaction between the actual deployment of schema.org and the definition, which is maintained in a community-driven fashion. The thesis has presented a novel, purely data-driven approach to measure the influence, which omits the necessity of a surveys. Based on the empirical findings, we can conclude that changes in the definition and the actual adoption affect each other. Furthermore, the thesis identified data consumers as the major drivers for an increasing deployment of certain classes and an increasing homogeneity of the adopted schema. This finding implies for data consumers, others than the search engine companies, the necessity of direct rewards for data providers in order to initiate the usage of a certain class or property.

Summarizing, in this part of the thesis, we have analyzed semantic annotations in terms of their general adoption in the Web, the topical coverage, as well as the quality of semantic annotations. In addition, the interactions between adoption and changes in the schema have been examined.

11.3 PART III: Use Case-Specific Profiling.

The chapters of the last part of the thesis have moved the focus towards an use case-specific analysis of the dataspace of semantic annotations.

In a first step, we have analyzed the utility of semantic annotations from schema.org Microdata dataspace to satisfy information needs, arising from actual use cases. We have shown, that the utility heavily depends on the use case and its requirements. Especially for more complex use cases, as the one of a travel portal, where detailed information about accommodation availability and prices are required, semantic annotations cannot satisfy this need. In contrast, for the use case

of a marketplace, the dataspace of semantic annotations are able to satisfy the use case-specific information need, at least the relevant semantic annotations made use of the corresponding properties (Section 9.4). The empirical findings underline the necessity of use case-specific profiling of the dataspace as well as the diversity of data availability across different topical groups of web sites.

In the last content chapter of the thesis, we have tried to create a more fine-grained topical distribution of product-related semantic annotations. In particular, we have tried to omit the usage of pre-trained classification models to enable an adaptability to other parts of the dataspace. Making use of a goldstandard including over 9 000 products from over 800 different web sites, the thesis has shown that solely distant-supervised learning approaches cannot compete with state-of-the-art supervised classification approaches. Also the refinement of the methods with background knowledge and the transformation to a global optimization problem, did not result in a comparable performance level.

Therefore, the creation of more fine-grained topical profiles for this dataspace requires manual-created or external training data, and hence depends on topic-specific knowledge and methods.

11.4 Research Impact

In this section, we discuss the impact of the contributions of this thesis for other researchers and studies. In general, we can only assess the impact based on (scientific) work which has been published. Hence, this assessment leaves out all work which has been done within companies and organizations, which do not provide their results and underlying methods to the public, including companies like Google.

11.4.1 Research Impact of Data Extraction Approaches

At the time of writing this thesis, no published scientific work has directly made use of the proposed focused crawling approach to harvest semantic annotations from the Web. Nevertheless, the extension which we provide for the *Apache Nutch* crawling framework has been forked (copied) over 780 times. Furthermore, ongoing work by Nutch contributors focuses on the integration of the proposed *selection strategy* within the standard Apache Nutch distribution.

The extraction framework (compare Chapter 4) was adapted by others to extract information from terabytes of web pages. The different research projects, exemplarily described in [Lehmberg et al., 2014b], have already been briefly outlined in Section 4.4. The diverse adaptions of the framework underline its usability for different use cases and research areas.

11.4.2 The Web Data Commons Project

As already stated in the thesis, all used corpora, intermediate results, and empirical findings, discussed in this thesis, have been published as part of the Web Data

Commons project. The project was initiated by Prof. Dr. Christian Bizer in 2012, focusing initially on the provision of semantic annotations embedded by Microformats, RDFa, and Microdata retrieved from the Common Crawl corpora. Since 2012, and as part of this thesis, four corpora containing data harvested in the years 2012, 2013, 2014, and 2015 have been made publicly available. The corpora are accompanied by the empirical findings and the software which is used to reproduce them. Besides the semantic annotations corpora, also other corpora, containing various information retrieved from the Common Crawl corpora, have been published as part of the Web Data Commons project. Within the first half of 2016 the project web site had around 5 000 visitors, where almost 1 000 visited the pages dedicated to the research presented in this thesis.[115]

11.4.3 Research Impact of Profiling Semantic Annotations

The impact of the work related to data provisioning and profiling is difficult to measure or assess. One could say, that the aggregated number of over 100 scientific publications, citing the findings presented in this thesis is a strong, positive indicator. Unfortunately, inspecting this selection of publications more closely, we found that a large fraction only mentions the existence of semantic annotations and the related findings, but still focus on their conventional data sources, e.g., the LOD cloud. In the last two years, some research started directly making use of the data and the findings presented in the publications related to this thesis.

In the following, we want to briefly mention scientific works from three different areas, making use of the data and the results provided as part of this thesis.

Scholarly Data The adoption of scholarly data is the major focus of the work presented in [Taibi and Dietze, 2016] and [Sahoo et al., 2016]. Using the semantic annotations corpora, extracted as in Chapter 4, together with the cleaning pipeline (compare Chapter 7), the authors create a more detailed profile of semantic annotations describing objects of the type `s:scholarlyArticle`. Beside a class and property based profiling, they found different bias in this part of the dataspace. First, most of the data providers are English and French. Second, the described topics of the articles mostly belong to the areas of computer science and life sciences. The analyses of the topical distribution and the overall adoption of semantic annotations (compare Chapter 6) are the foundation for their work.

Tourism The influence of semantic annotations in HTML pages, embedded by Microformats, RDFa, and Microdata for the tourism sector has been studied in [Stavrakantonakis et al., 2013] and [Kärle et al., 2016]. The corpora extracted in Chapter 4 are the data foundation for their analyses. In their work, the authors identified a gap between the technological possibilities to provide data and improve the visibility in the Web (e.g., through search engine results) and the actual used

[115] The reported numbers relate to unique visitors and were collected using *Google Analytics*.

technologies by web sites about hotels in Austria. They also discovered data quality issues based on geographical information contained in semantic annotations. The main motivation of their work is more business-related, and focuses on the enabling of Austrian tourism to benefit from the idea of the semantic web.

Products In the area of product-related semantic annotations, the research presented in [Petrovski et al., 2014] has especially focused on the problems arising from the shallowness of described product-related entities, based on the findings of Chapter 6 and Chapter 9. The authors identified that parts of the title and descriptions potentially include additional attributes. Exemplary, product titles like "apple iPod 64GB black" include other useful attributes. Beside the brand (apple), also the color (black) and the size of the internal storage (64GB) is contained. As web site do not mark those information individually, they have to be extracted separately. The group around the authors focus on the identification of approaches to extract those contained attributes in product-related semantic annotations. Furthermore, they focused in their recent work [Petrovski et al., 2016] on the combination of semantic annotations from Microdata and data contained in HTML tables and lists, to generate comprehensive product descriptions.

A work, focusing on the enrichment of product advertisements with information contained in semantic annotations is presented in [Ristoski and Mika, 2016]. The authors make use of the semantic annotations extracted as part of this thesis and evaluate their approach based on the goldstandard presented in Chapter 10. Based on our findings with respect to the usage of distant-supervised approaches for the classification of products-related semantic annotations, they again make use of a supervised approach for product feature extraction as well as product classification.

Summarizing, we think that semantic annotations will become more and more interesting in the next years. We believe that by the findings and approaches presented in this thesis, we have created a solid foundation for future research focusing on various aspects of semantic annotations in HTML pages.

List of Figures

List of Tables

Listings

Bibliography

[Abedjan et al., 2014] Abedjan, Z., Gruetze, T., Jentzsch, A., and Naumann, F. (2014). Profiling and mining rdf data with prolod++. In *Proceedings of the 2014 IEEE 30th International Conference on Data Engineering*, ICDE'14, pages 1198–1201.

[Abedjan et al., 2012] Abedjan, Z., Lorey, J., and Naumann, F. (2012). Reconciling ontologies and the web of data. In *Proceedings of the 21st International Conference on Information and Knowledge Management*, CIKM '12, pages 1532–1536, Maui, Hawaii, USA.

[Adida and Birbeck, 2008] Adida, B. and Birbeck, M. (2008). *RDFa Primer - Bridging the Human and Data Webs - W3C Recommendation*.

[Aggarwal et al., 2001] Aggarwal, C. C., Al-Garawi, F., and Yu, P. S. (2001). Intelligent crawling on the world wide web with arbitrary predicates. In *Proceedings of the 20th International Conference on World Wide Web*, WWW'01, New York, NY, USA.

[Ashraf et al., 2011] Ashraf, J., Cyganiak, R., O'Riain, S., and Hadzic, M. (2011). Open ebusiness ontology usage: Investigating community implementation of goodrelations. In *Proceedings of the WWW2011 Workshop on Linked Data on the Web*, LDOW WWW'11.

[Baeza-Yates and Poblete, 2003] Baeza-Yates, R. and Poblete, B. (2003). Evolution of the Chilean web structure composition. In *Proceedings of the of Latin American Web Conference 2003*, LA-WEB'03, pages 11–13.

[Baeza-Yates et al., 1999] Baeza-Yates, R., Ribeiro-Neto, B., et al. (1999). *Modern information retrieval*, volume 463. ACM Press.

[Bakalov et al., 2011] Bakalov, A., Fuxman, A., Talukdar, P. P., and Chakrabarti, S. (2011). Scad: Collective discovery of attribute values. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 447–456, New York, NY, USA. ACM.

[Barbosa and Freire, 2007] Barbosa, L. and Freire, J. (2007). An adaptive crawler for locating hidden-web entry points. In *Proceedings of the 16th International Conference on World Wide Web*, WWW'07, New York, NY, USA.

[Beek et al., 2014] Beek, W., Rietveld, L., Bazoobandi, H. R., Wielemaker, J., and Schlobach, S. (2014). Lod laundromat: A uniform way of publishing other people's dirty data. In *Proceedings of the 13th International Semantic Web Conference*, ISWC'14. Springer.

[Berners-Lee, 2006] Berners-Lee, T. (2006). Linked data. `https://www.w3.org/DesignIssues/LinkedData.html`.

[Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., Lassila, O., and othfers (2001). The semantic web. *Scientific American*, 284(5):28–37.

[Bifet et al., 2010] Bifet, A., Holmes, G., Kirkby, R., and Pfahringer, B. (2010). MOA: Massive online analysis. *Journal of Machine Learning Research*, 11(May):1601–1604.

[Bizer et al., 2013] Bizer, C., Eckert, K., Meusel, R., Mühleisen, H., Schuhmacher, M., and Völker, J. (2013). Deployment of rdfa, microdata, and microformats on the web – a quantitative analysis. In *Proceedings of the 12th International Semantic Web Conference*, ISWC'13. Springer Berlin Heidelberg.

[Bizer et al., 2009] Bizer, C., Heath, T., and Berners-Lee, T. (2009). Linked data-the story so far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, 1:205–227.

[Bohannon et al., 2005] Bohannon, P., Fan, W., Flaster, M., and Rastogi, R. (2005). A cost-based model and effective heuristic for repairing constraints by value modification. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, SIGMOD'05, pages 143–154, New York, NY, USA. ACM.

[Boldi et al., 2002] Boldi, P., Codenotti, B., Santini, M., and Vigna, S. (2002). Structural properties of the African web. In *Proceedings of the 11th International Conference on World Wide Web*, WWW'02.

[Boldi et al., 2004] Boldi, P., Codenotti, B., Santini, M., and Vigna, S. (2004). UbiCrawler: a scalable fully distributed Web crawler. *Software-Practice and Experience*, 34(8):711–726.

[Breslin et al., 2005] Breslin, J., Harth, A., Bojars, U., and Decker, S. (2005). Towards semantically-interlinked online communities. In *Proceedings of the 2nd European Semantic Web Conference*, ESWC'05, Heraklion, Greece.

[Brickley, 2015] Brickley, D. (2015). Schema.org hierarchy sunburst. Blog Post: `http://bl.ocks.org/danbri/1c121ea8bd2189cf411c`.

[Brickley, 2016] Brickley, D. (2016). GS1 Web vocabulary: welcoming the first schema.org external extension. `http://blog.schema.org/2016/02/gs1-milestone-first-schemaorg-external.html`.

[Broder et al., 2000] Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., and Wiener, J. (2000). Graph structure in the Web: experiments and models. *Computer Networks*, 33(1–6):309–320.

[Bryl et al., 2015] Bryl, V., Bizer, C., and Paulheim, H. (2015). Gathering alternative surface forms for dbpedia entities. In *Proceedings of the 3rd Workshop on NLP & DBpedia*, NLP & DBpedia'15.

[Bugiotti et al., 2012] Bugiotti, F., Goasdoué, F., Kaoudi, Z., and Manolescu, I. (2012). Rdf data management in the amazon cloud. In *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, EDBT-ICDT'12, pages 61–72, New York, NY, USA. ACM.

[Cafarella et al., 2008] Cafarella, M. J., Halevy, A., Wang, D. Z., Wu, E., and Zhang, Y. (2008). Webtables: Exploring the power of tables on the web. *Proceedings of the VLDB Endowment*, 1(1):538–549.

[Cambazoglu and Baeza-Yates, 2011] Cambazoglu, B. B. and Baeza-Yates, R. (2011). Scalability challenges in web search engines. In *Advanced topics in information retrieval*, pages 27–50. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Cambazoglu and Baeza-Yates, 2015] Cambazoglu, B. B. and Baeza-Yates, R. (2015). Scalability challenges in web search engines. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 7(6):1–138.

[Chakrabarti et al., 2002a] Chakrabarti, S., Joshi, M. M., Punera, K., and Pennock, D. M. (2002a). The structure of broad topics on the web. In *Proceedings of the 11th international conference on World Wide Web*, WWW'02, pages 251–262. ACM.

[Chakrabarti et al., 2002b] Chakrabarti, S., Punera, K., and Subramanyam, M. (2002b). Accelerated focused crawling through online relevance feedback. In *Proceedings of the 11th international conference on World Wide Web*, WWW'02, New York, NY, USA.

[Chakrabarti et al., 1999] Chakrabarti, S., van den Berg, M., and Dom, B. (1999). Focused crawling: New approach to topic-specific web resource discovery. *Computer Networks*, 31(11):1623–1640.

[Chapelle and Li, 2011] Chapelle, O. and Li, L. (2011). An empirical evaluation of thompson sampling. In *Proceedings of the 25th Conference on Neural Information Processing Systems*, NIPS'11.

[Chau and Hui, 2001] Chau, P. Y. and Hui, K. L. (2001). Determinants of small business edi adoption: an empirical investigation. *Journal of Organizational Computing and Electronic Commerce*, 11(4):229–252.

[Chen, 2003] Chen, M. (2003). Factors affecting the adoption and diffusion of xml and web services standards for e-business systems. *International Journal of Human-Computer Studies*, 58(3):259–279.

[Chen, 2005] Chen, M. (2005). An analysis of the driving forces for web services adoption. *Information Systems and e-Business Management*, 3(3):265–279.

[Chen et al., 2005] Chen, S., Hong, D., and Shen, V. (2005). An experimental study on validation problems with existing html webpages. In *Proceedings of the 2005 International Conference on Internet Computing, ICOMP'05*.

[Chiao et al., 2007] Chiao, B., Lerner, J., and Tirole, J. (2007). The rules of standard-setting organizations: an empirical analysis. *The RAND Journal of Economics*, 38(4):905–930.

[Chu et al., 2013] Chu, X., Ilyas, I., and Papotti, P. (2013). Holistic data cleaning: Putting violations into context. In *Proceedings of the 29th International Conference on Data Engineering*, ICDE'13, pages 458–469.

[Chu et al., 2015] Chu, X., Morcos, J., Ilyas, I. F., Ouzzani, M., Papotti, P., Tang, N., and Ye, Y. (2015). Katara: A data cleaning system powered by knowledge bases and crowdsourcing. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, pages 1247–1261, New York, NY, USA. ACM.

[Ciganek et al., 2006] Ciganek, A. P., Haines, M. N., and Haseman, W. (2006). Horizontal and vertical factors influencing the adoption of web services. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences*, volume 6 of *HICSS'06*, pages 109c–109c. IEEE.

[Clauset et al., 2009] Clauset, A., Shalizi, C. R., and Newman, M. E. J. (2009). Power-law distributions in empirical data. *SIAM Review*, 51(4):661–703.

[Cutrell and Guan, 2007] Cutrell, E. and Guan, Z. (2007). What are you looking for?: An eye-tracking study of information usage in web search. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI'07, pages 407–416, New York, NY, USA. ACM.

[Dallachiesa et al., 2013] Dallachiesa, M., Ebaid, A., Eldawy, A., Elmagarmid, A., Ilyas, I. F., Ouzzani, M., and Tang, N. (2013). Nadeef: A commodity data cleaning system. In *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, SIGMOD'13, pages 541–552, New York, NY, USA. ACM.

[Dasgupta et al., 2007] Dasgupta, A., Ghosh, A., Kumar, R., Olston, C., Pandey, S., and Tomkins, A. (2007). The discoverability of the web. In *Proceedings of the 16th international conference on World Wide Web*, WWW'07. ACM.

[Dhenakaran and Sambanthan, 2011] Dhenakaran, S. and Sambanthan, K. T. (2011). Web crawler–an overview. *International Journal of Computer Science and Communication*, 2:265–267.

[Diligenti et al., 2000] Diligenti, M., Coetzee, F., Lawrence, S., Giles, C. L., Gori, M., et al. (2000). Focused crawling using context graphs. *Proceedings of the VLDB Endowment*.

[Ding et al., 2002] Ding, Y., Korotkiy, M., Omelayenko, B., Kartseva, V., Zykov, V., Klein, M., Schulten, E., and Fensel, D. (2002). Goldenbullet: Automated classification of product data in e-commerce. In *Proceedings of the 5th International Conference on Business Information Systems*.

[Dixon, 2010] Dixon, J. (2010). Pentaho, hadoop, and data lakes. Blog Post: `https://jamesdixon.wordpress.com/2010/10/14/pentaho-hadoop-and-data-lakes/`.

[Doan et al., 2012] Doan, A., Halevy, A., and Ives, Z. (2012). *Principles of data integration*. Elsevier.

[Dodds, 2006] Dodds, L. (2006). Slug: A semantic web crawler. In *Proceedings of the Jena User Conference*.

[Domingos and Lowd, 2009] Domingos, P. and Lowd, D. (2009). Markov logic: An interface layer for artificial intelligence. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 3(1):1–155.

[Donato et al., 2005] Donato, D., Leonardi, S., Millozzi, S., and Tsaparas, P. (2005). Mining the inner structure of the web graph. In *Proceedings of the 8th International Workshop on the Web & Databases*, WebDB'05, pages 145–150.

[Dorogovtsev et al., 2001] Dorogovtsev, S. N., Mendes, J. F. F., and Samukhin, A. N. (2001). Giant strongly connected component of directed networks. *Physical Review E*, 64:025101.

[eMarketer, 2014] eMarketer (2014). Worldwide ecommerce sales to increase nearly 20% in 2014. `http://www.emarketer.com/Article/Worldwide-Ecommerce-Sales-Increase-Nearly-20-2014/1011039`.

[eMarketer, 2015] eMarketer (2015). Worldwide retail ecommerce sales: emarketer's updated estimates and forecast through 2019. `http://www.emarketer.com/Report/Worldwide-Retail-Ecommerce-Sales-eMarketers- Updated-Estimates-Forecast-Through-2019/2001716`.

[Fan et al., 2013] Fan, W., Geerts, F., Tang, N., and Yu, W. (2013). Inferring data currency and consistency for conflict resolution. In *Proceedings of the 29th International Conference on Data Engineering*, ICDE'13, pages 470–481.

[Fan et al., 2014] Fan, W., Ma, S., Tang, N., and Yu, W. (2014). Interaction between record matching and data repairing. *ACM Journal of Data and Information Quality*, 4(4):16:1–16:38.

[Fawcett, 2006] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, 27(8):861–874.

[Franklin et al., 2005] Franklin, M., Halevy, A., and Maier, D. (2005). From databases to dataspaces: A new abstraction for information management. *ACM SIGMOD Record*, 34(4):27–33.

[Geerts et al., 2013] Geerts, F., Mecca, G., Papotti, P., and Santoro, D. (2013). The llunatic data-cleaning framework. *Proceedings of the VLDB Endowment*, 6(9):625–636.

[Glimm et al., 2012] Glimm, B., Hogan, A., Krötzsch, M., and Polleres, A. (2012). OWL: yet to arrive on the web of data? In *Proceedings of the WWW2012 Workshop on Linked Data on the Web*, LDOW WWW'12, pages 1–10.

[Goel et al., 2009] Goel, K., Guha, R. V., and Hansson, O. (2009). Introducing rich snippets. `http://googlewebmastercentral.blogspot.de/2009/05/introducing-rich-snippets.html`.

[Gonina et al., 2011] Gonina, E., Kannan, A., Shafer, J., and Budiu, M. (2011). Parallelizing large-scale data processing applications with data skew: A case study in product-offer matching. In *Proceedings of the Second International Workshop on MapReduce and Its Applications*, MapReduce '11, pages 35–42, New York, NY, USA. ACM.

[Google Inc., 2015] Google Inc. (2015). Structured data - rich snippets. https://developers.google.com/structured-data/rich-snippets/.

[Graves et al., 2007] Graves, M., Constabaris, A., and Brickley, D. (2007). Foaf: Connecting people on the semantic web. *Cataloging & classification quarterly*, 43(3-4):191–202.

[Guha et al., 2015] Guha, R., Brickley, D., and Macbeth, S. (2015). Schema.org: Evolution of structured data on the web. *acmqueue*, 13(9):1–28.

[Guha, 2014] Guha, R. V. (2014). Schema.org update. `http://events.linkeddata.org/ldow2014/slides/ldow2014_keynote_guha_schema_org.pdf`.

[Gummer, 2015] Gummer, T. (2015). *Multiple Panels in der empirischen Sozialforschung: Evaluation eines Forschungsdesigns mit Beispielen aus der Wahlsoziologie*. Springer VS, Wiesbaden.

[Han and Pei, 2000] Han, J. and Pei, J. (2000). Mining frequent patterns by pattern-growth: Methodology and implications. *SIGKDD Exploration Newsletter*, 2(2):14–20.

[Harth et al., 2009] Harth, A., Kinsella, S., and Decker, S. (2009). Using naming authority to rank data and ontologies for web search. In *Proceedings of the 8th International Semantic Web Conference*, ISWC'09.

[Hepp, 2008] Hepp, M. (2008). Goodrelations: An ontology for describing products and services offers on the web. In *Proceedings of the 16th International Conference on Knowledge Engineering: Practice and Patterns.*, EKAW'08, pages 329–346, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Hernández and Stolfo, 1998] Hernández, M. A. and Stolfo, S. J. (1998). Real-world data is dirty: Data cleansing and the merge/purge problem. *Data Mining and Knowledge Discovery*, 2(1):9–37.

[Hickson, 2011] Hickson, I. (2011). HTML Microdata. `http://www.w3.org/TR/microdata/`. Working Draft.

[Hickson et al., 2014] Hickson, I., Kellogg, G., Tennison, J., and Herman, I. (2014). Microdata to rdf – second edition. `http://www.w3.org/TR/microdata-rdf/`.

[Hogan et al., 2010a] Hogan, A., Harth, A., Passant, A., Decker, S., and Polleres, A. (2010a). Weaving the pedantic web. In *Proceedings of the WWW2010 Workshop on Linked Data on the Web*, LDOW WWW'10.

[Hogan et al., 2010b] Hogan, A., Polleres, A., Umbrich, J., and Zimmermann, A. (2010b). Some entities are more equal than others: statistical methods to consolidate linked data. In *Proceedings of the 4th International Workshop on New Forms of Reasoning for the Semantic Web: Scalable and Dynamic*, NeFoRS'10.

[Hogan et al., 2012] Hogan, A., Umbrich, J., Harth, A., Cyganiak, R., Polleres, A., and Decker, S. (2012). An empirical survey of linked data conformance. *Web Semantics: Science, Services and Agents on the World Wide Web*, 14:14 – 44. Special Issue on Dealing with the Messiness of the Web of Data.

[Horvitz and Thompson, 1952] Horvitz, D. G. and Thompson, D. J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260):663–685.

[Ilyas and Chu, 2012] Ilyas, I. F. and Chu, X. (2012). Trends in cleaning relational data: Consistency and deduplication. *Foundations and Trends in Databases*, 5(4):281–393.

[Isele et al., 2010] Isele, R., Umbrich, J., Bizer, C., and Harth, A. (2010). LDSpider: An open-source crawling framework for the web of linked data. In *Proceedings of the ISWC'10 Posters & Demonstrations Track: Collected Abstracts*.

[Jiang et al., 2012] Jiang, J., Yu, N., and Lin, C.-Y. (2012). Focus: Learning to crawl web forums. In *Proceedings of the 21th international conference on World Wide Web*, WWW'12, New York, USA. ACM.

[Kan, 2004] Kan, M.-Y. (2004). Web page classification without the web page. In *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*, WWW Alt.'04, New York, NY, USA. ACM.

[Kan and Thi, 2005] Kan, M.-Y. and Thi, H. O. N. (2005). Fast webpage classification using url features. In *Proceedings of the 14th International Conference on Information and Knowledge Management*, CIKM'05.

[Kane and Alavi, 2007] Kane, G. and Alavi, M. (2007). Information technology and organizational learning: An investigation of exploration and exploitation processes. *Orginazation Science*, 18(5):796–812.

[Kannan et al., 2011a] Kannan, A., Givoni, I., Agrawal, R., and Fuxman, A. (2011a). Matching unstructured offers to structured product descriptions. In *International Conference on Knowledge Discovery and Data Mining*, KDD'11. ACM.

[Kannan et al., 2011b] Kannan, A., Talukdar, P., Rasiwasia, N., and Ke, Q. (2011b). Improving product classification using images. In *Proceedings of the 11th International Conference on Data Mining*, ICDM'11, pages 310–319.

[Kärle et al., 2016] Kärle, E., Fensel, A., Toma, I., and Fensel, D. (2016). Why are there more hotels in tyrol than in austria? analyzing schema.org usage in the hotel domain. In *Proceedings of the International Conference of Information and Communication Technologies in Tourism*, pages 99–112, Cham. Springer International Publishing.

[Khare et al., 2004] Khare, R., Cutting, D., Sitaker, K., and Rifkin, A. (2004). Nutch: A flexible and scalable open-source web search engine. *Oregon State University*, 32:1–12.

[Klyne and Carroll, 2004] Klyne, G. and Carroll, J. J. (2004). *Resource Description Framework (RDF): Concepts and Abstract Syntax - W3C Recommendation*. http://www.w3.org/TR/rdf-concepts/.

[Kolahi and Lakshmanan, 2009] Kolahi, S. and Lakshmanan, L. V. S. (2009). On approximating optimum repairs for functional dependency violations. In *Proceedings of the 12th International Conference on Database Theory*, ICDT'09, pages 53–62, New York, NY, USA. ACM.

[Kolb, 2008] Kolb, P. (2008). Disco: A multilingual database of distributionally similar words. In *In Proceedings of the 9th Konferenz zur Verarbeitung natürlicher Sprache*, Berlin, Germany.

[Köpcke et al., 2012] Köpcke, H., Thor, A., Thomas, S., and Rahm, E. (2012). Tailoring entity resolution for matching product offers. In *Proceedings of the 15th International Conference on Extending Database Technology*, EDBT'12, pages 545–550, New York, NY, USA. ACM.

[Kowalczuk et al., 2014] Kowalczuk, E., Potoniec, J., and Lawrynowicz, A. (2014). Extracting usage patterns of ontologies on the web: a case study on goodrelations vocabulary in rdfa. In *Proceedings of the ISWC2014 Workshop OWL: Experiences and Directions*, OWLED'14, pages 139–144.

[Kruskal and Mosteller, 1979a] Kruskal, W. and Mosteller, F. (1979a). Representative sampling, i: Non-scientific literature. *International Statistical Review/Revue Internationale de Statistique*, 47(1):13–24.

[Kruskal and Mosteller, 1979b] Kruskal, W. and Mosteller, F. (1979b). Representative sampling, ii: Scientific literature, excluding statistics. *International Statistical Review/Revue Internationale de Statistique*, 47(2):111–127.

[Kruskal and Mosteller, 1979c] Kruskal, W. and Mosteller, F. (1979c). Representative sampling, iii: The current statistical literature. *International Statistical Review/Revue Internationale de Statistique*, 47(3):245–265.

[Kutuzov and Ionov, 2014] Kutuzov, A. and Ionov, M. (2014). *Untangling the Semantic Web: Microdata Use in Russian Video Content Delivery Sites*, pages 274–279. Springer International Publishing, Cham.

[Lehmberg et al., 2014a] Lehmberg, O., Meusel, R., and Bizer, C. (2014a). Graph structure in the web: aggregated by pay-level domain. In *Proceedings of the 2014 ACM conference on Web Science*, WebSci'14, pages 119–128. ACM.

[Lehmberg et al., 2014b] Lehmberg, O., Ritze, D., Ristoski, P., Eckert, K., Paulheim, H., and Bizer, C. (2014b). Extending tables with data from over a million websites. In *Semantic Web Challenge*.

[Leskovec and Faloutsos, 2006] Leskovec, J. and Faloutsos, C. (2006). Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD'06, pages 631–636, New York, NY, USA. ACM.

[Li et al., 2010] Li, L., Chu, W., Langford, J., and Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW'10, pages 661–670, New York, NY, USA. ACM.

[Lindahl, 2012] Lindahl, G. (2012). Blekko donates search data to common crawl. `http://blog.blekko.com/2012/12/17/common-crawl-donation/`.

[Lohr, 1999] Lohr, S. L. (1999). *Sampling: Design and Analysis*. Duxbury Press.

[Mannino et al., 1988] Mannino, M. V., Chu, P., and Sager, T. (1988). Statistical profile estimation in database systems. *ACM Computing Surveys*, 20(3):191–221.

[March, 1991] March, J. G. (1991). Exploration and exploitation in organizational learning. *Orginazation Science*, 2(1):71–87.

[Menczer, 1997] Menczer, F. (1997). ARACHNID: Adaptive Retrieval Agents Choosing Heuristic Neighborhoods for Information Discovery. In *Proceedings of the 28th International Conference on Machine Learning*, ICML'11, San Francisco, CA. Morgan Kaufmann.

[Meusel et al., 2015a] Meusel, R., Bizer, C., and Paulheim, H. (2015a). A web-scale study of the adoption and evolution of the schema. org vocabulary over time. In *Proceedings of the 5th International Conference on Web Intelligence, Mining and Semantics*, WIMS'15, pages 15:1–15:11, New York, NY, USA. ACM, ACM.

[Meusel et al., 2014a] Meusel, R., Mika, P., and Blanco, R. (2014a). Focused crawling for structured data. In *Proceedings of the 23rd Conference on Information and Knowledge Management*, CIKM'14, New York, NY, USA. ACM.

[Meusel and Paulheim, 2015] Meusel, R. and Paulheim, H. (2015). Heuristics for fixing common errors in deployed schema.org microdata. In *Proceedings of the 12th European Semantic Web Conference*, ESWC'15, pages 152–168, Cham. Springer International Publishing.

[Meusel et al., 2014b] Meusel, R., Petrovski, P., and Bizer, C. (2014b). The web-datacommons microdata, rdfa and microformat dataset series. In *Proceedings of the 13th International Semantic Web Conference*, ISWC'14, pages 277–292. Springer International Publishing.

[Meusel et al., 2015b] Meusel, R., Primpeli, A., Meilicke, C., Paulheim, H., and Bizer, C. (2015b). Exploiting microdata annotations to consistently categorize product offers at web scale. In *Proceedings of the 16th International Conference on Electronic Commerce and Web Technologies*, EC-Web'15, pages 83–99, Cham. Springer International Publishing.

[Meusel et al., 2016] Meusel, R., Ritze, D., and Paulheim, H. (2016). Towards more accurate statistical profiling of deployed schema.org microdata. *J. Data and Information Quality*, 8(1):3:1–3:31.

[Meusel et al., 2014c] Meusel, R., Vigna, S., Lehmberg, O., and Bizer, C. (2014c). Graph structure in the web - revisited: a trick of the heavy tail. In *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, WWW Comp.'14, pages 427–432. International World Wide Web Conferences Steering.

[Meusel et al., 2014d] Meusel, R., Vigna, S., Lehmberg, O., and Bizer, C. (2014d). Topology of the 2014 wdc hyperlink graph. Website: `http://webdatacommons.org/hyperlinkgraph/2014-04/topology.html`.

[Meusel et al., 2015c] Meusel, R., Vigna, S., Lehmberg, O., and Bizer, C. (2015c). The graph structure in the web–analyzed on different aggregation levels. *The Journal of Web Science*, 1(1):33–47.

[Mika, 2011] Mika, P. (2011). Microformats and RDFa deployment across the Web . `http://tripletalk.wordpress.com/2011/01/25/rdfa-deployment-across-the-web/`.

[Mika and Potter, 2012] Mika, P. and Potter, T. (2012). Metadata statistics for a large web corpus. In *Proceedings of the WWW2012 Workshop on Linked Data on the Web*, LDOW WWW'12. CEUR-ws.org.

[Miles et al., 2005] Miles, A., Matthews, B., Wilson, M., and Brickley, D. (2005). Skos core: Simple knowledge organisation for the web. In *Proceedings of the 2005 International Conference on Dublin Core and Metadata Applications: Vocabularies in Practice*, DCMI '05, pages 1:1–1:9. Dublin Core Metadata Initiative.

[Mirkovic and Reiher, 2004] Mirkovic, J. and Reiher, P. (2004). A taxonomy of ddos attack and ddos defense mechanisms. *SIGCOMM Computer Communication Review*, 34(2):39–53.

[Moreno-Torres et al., 2012] Moreno-Torres, J. G., Raeder, T., Alaiz-Rodríguez, R., Chawla, N. V., and Herrera, F. (2012). A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530.

[Mühleisen and Bizer, 2012] Mühleisen, H. and Bizer, C. (2012). Web data commons – extracting structured data from two large web corpora. In *Proceedings of the WWW2012 Workshop on Linked Data on the Web*, LDOW WWW'12. CEUR-ws.org.

[Naumann, 2014] Naumann, F. (2014). Data profiling revisited. *ACM SIGMOD Record*, 42(4):40–49.

[Nguyen et al., 2011] Nguyen, H., Fuxman, A., Paparizos, S., Freire, J., and Agrawal, R. (2011). Synthesizing products for online catalogs. *Proceedings of the VLDB Endowment*, 4(7):409–418.

[Noessner et al., 2013] Noessner, J., Niepert, M., and Stuckenschmidt, H. (2013). Rockit: Exploiting parallelism and symmetry for MAP inference in statistical relational models. In *Proceedings of the 27th Conference on Artificial Intelligence*, AAAI'13.

[Olston and Najork, 2010] Olston, C. and Najork, M. (2010). Web crawling. *Foundations and Trends in Information Retrieval*, 4(3):175–246.

[Ozsu, 2007] Ozsu, M. T. (2007). *Principles of Distributed Database Systems*. Prentice Hall Press, Upper Saddle River, NJ, USA, 3rd edition.

[Page et al., 1999] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab.

[Pandey et al., 2007] Pandey, S., Chakrabarti, D., and Agarwal, D. (2007). Multi-armed bandit problems with dependent arms. In *Proceedings of the 24th International Conference on Machine learning*, ICML'07, pages 721–728, New York, NY, USA. ACM.

[Pant et al., 2002] Pant, G., Srinivasan, P., Menczer, F., et al. (2002). Exploration versus exploitation in topic driven crawlers. In *Proceedings of the WWW2002 Workshop on Web Dynamics*, WebDyn WWW'02. Citeseer.

[Papadimitriou et al., 2012] Papadimitriou, P., Tsaparas, P., Fuxman, A., and Getoor, L. (2012). Taci: Taxonomy-aware catalog integration. *IEEE Transactions on Knowledge and Data Engineering*, 25:1643–1655.

[Papenbrock et al., 2015] Papenbrock, T., Bergmann, T., Finke, M., Zwiener, J., and Naumann, F. (2015). Data profiling with metanome. *Proceedings of the VLDB Endow.*, 8(12):1860–1863.

[Patel-Schneider, 2014] Patel-Schneider, P. F. (2014). Analyzing Schema.org. In *Proceedings of the 13th International Semantic Web Conference*, ISWC'14, Cham. Springer International Publishing.

[Paulheim, 2015] Paulheim, H. (2015). What the adoption of schema.org tells about linked open data. In *Proceedings of the 2nd International Workshop on Dataset PROFIling & fEderated Search for Linked Data*, USEWOD-PROFILES ESWC'15.

[Petrovski et al., 2014] Petrovski, P., Bryl, V., and Bizer, C. (2014). Integrating product data from websites offering microdata markup. In *Proceedings of the 4th Workshop on Data Extraction and Object Search*, DEOS WWW'14.

[Petrovski et al., 2016] Petrovski, P., Primpeli, A., Meusel, R., and Bizer, C. (2016). The WDC gold standard for product feature extraction and product matching. In *Proceedings of 17th International Conference on Electronic Commerce and Web Technologies*. accepted.

[Porter, 1980] Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

[Prud'Hommeaux et al., 2008] Prud'Hommeaux, E., Seaborne, A., et al. (2008). Sparql query language for rdf. *W3C recommendation*, 15.

[Puurula, 2012] Puurula, A. (2012). Scalable text classification with sparse generative modeling. In *Proceedings of the Pacific Rim International Conference on Artifical Intelligence*, PRICAI'12. Springer.

[Qiu et al., 2015] Qiu, D., Barbosa, L., Dong, X. L., Shen, Y., and Srivastava, D. (2015). Dexter: Large-scale discovery and extraction of product specifications on the web. *Proceedings of the VLDB Endowment*, 8(13):2194–2205.

[Rheinländer et al., 2016] Rheinländer, A., Lehmann, M., Kunkel, A., Meier, J., and Leser, U. (2016). Potential and pitfalls of domain-specific information extraction at web scale. In *Proceedings of the 2016 ACM SIGMOD International Conference on Management of Data*, SIGMOD'16. ACM.

[Ristoski et al., 2014] Ristoski, P., Mencía, E. L., and Paulheim, H. (2014). A hybrid multi-strategy recommender system using linked open data. In *Semantic Web Evaluation Challenge*, pages 150–156. Springer International Publishing.

[Ristoski and Mika, 2016] Ristoski, P. and Mika, P. (2016). Enriching product ads with metadata from html annotations. In *Proceedings of the 15th International Semantic Web Conference*, ISWC'16, pages 151–167. Springer.

[Ristoski et al., 2015] Ristoski, P., Mika, P., Blanco, R., and Meusel, R. (2015). Explore anthelion, our open source focused crawler. `http://yahoolabs.tumblr.com/post/135196452221/explore-anthelion-our-open-source-focused-crawler`.

[Sahoo et al., 2016] Sahoo, P., Gadiraju, U., Yu, R., Saha, S., and Dietze, S. (2016). Analysing structured scholarly data embedded in web pages. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW Comp.'16.

[Schmachtenberg et al., 2014] Schmachtenberg, M., Bizer, C., and Paulheim, H. (2014). Adoption of the linked data best practices in different topical domains. In *Proceedings of the 13th International Semantic Web Conference*, ISWC'14, Cham. Springer International Publishing.

[Schnell et al., 2011] Schnell, R., Hill, P. B., and Esser, E. (2011). *Methoden der empirischen Sozialforschung*. Oldenbourg Verlag.

[Seitner et al., 2016] Seitner, J., Bizer, C., Eckert, K., Faralli, S., Meusel, R., Paulheim, H., and Ponzetto, S. (2016). A large database of hypernymy relations extracted from the web. *Proceedings of the 10th Edition of the Language Resources and Evaluation Conference*.

[Serrano et al., 2007] Serrano, M., Maguitman, A., Boguñá, M., Fortunato, S., and Vespignani, A. (2007). Decoding the structure of the WWW: A comparative analysis of web crawls. *ACM Transactions on the Web*, 1(2):10.

[Shannon, 2001] Shannon, C. E. (2001). A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1):3–55.

[Shi et al., 2009] Shi, Q., Petterson, J., Dror, G., Langford, J., Smola, A., and Vishwanathan, S. (2009). Hash kernels for structured data. *The Journal of Machine Learning Research*, 10:2615–2637.

[Spiegler, 2013] Spiegler, S. (2013). Statistcs of the common crawl corpus 2012. Technical report, SwiftKey.

[Stavrakantonakis et al., 2013] Stavrakantonakis, I., Toma, I., Fensel, A., and Fensel, D. (2013). Hotel websites, web 2.0, web 3.0 and online direct marketing: The case of austria. In Xiang, Z. and Tussyadiah, I., editors, *Information and Communication Technologies in Tourism 2014*, pages 665–677. Springer International Publishing.

[Stolz and Hepp, 2015] Stolz, A. and Hepp, M. (2015). Towards crawling the web for structured data: Pitfalls of common crawl for e-commerce. In *Proceedings of the 6th International Workshop on Consuming Linked Data*, COLD ISWC'15. CEUR-ws.org.

[Taibi and Dietze, 2016] Taibi, D. and Dietze, S. (2016). Towards embedded markup of learning resources on the web: An initial quantitative analysis of lrmi terms usage. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW Comp.'16, pages 513–517, Republic and Canton of Geneva, Switzerland. International World Wide Web Conferences Steering Committee.

[Toma et al., 2014] Toma, I., Stanciu, C., Fensel, A., Stavrakantonakis, I., and Fensel, D. (2014). Improving the online visibility of touristic service providers by using semantic annotations. In *Proceedings of the 11th Extended Semantic Web Conference: Satellite Events*, volume 8798 of *ESWC'14*, pages 259–262. Springer International Publishing.

[Udrea et al., 2007] Udrea, O., Getoor, L., and Miller, R. J. (2007). Leveraging data and structure in ontology integration. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD'07, pages 449–460, New York, NY, USA. ACM.

[Umbrich et al., 2009] Umbrich, J., Karnstedt, M., and Harth, A. (2009). Fast and scalable pattern mining for media-type focused crawling. *Proceedings of the Workshop on Knowledge Discovery, Data Mining and Machine Learning*, 1:119–126.

[Vaisman and Zimnyi, 2014] Vaisman, A. and Zimnyi, E. (2014). *Data Warehouse Systems*. Springer Berlin Heidelberg.

[Weibel, 1997] Weibel, S. (1997). The dublin core: a simple content description model for electronic resources. *Bulletin of the American Society for Information Science and Technology*, 24(1):9–11.

[White, 2012] White, T. (2012). *Hadoop: The definitive guide*. " O'Reilly Media, Inc.".

[Yee-Loong Chong and Ooi, 2008] Yee-Loong Chong, A. and Ooi, K.-B. (2008). Adoption of interorganizational system standards in supply chains: an empirical analysis of rosettanet standards. *Industrial Management & Data Systems*, 108(4):529–547.

[Yu et al., 2015] Yu, R., Gadiraju, U., Fetahu, B., and Dietze, S. (2015). Adaptive focused crawling of linked data. In Wang, J., Cellary, W., Wang, D., Wang, H., Chen, S.-C., Li, T., and Zhang, Y., editors, *Proceedings of the 16th Internatinoal Conference on Web Information Systems Engineering*, volume 9418 of *WISE'15*, pages 554–569. Springer International Publishing.

[Zaveri et al., 2015] Zaveri, A., Rula, A., Maurino, A., Pietrobon, R., Lehmann, J., and Auer, S. (2015). Quality assessment methodologies for linked data: A survey. *Semantic Web*, 7(1):63–93.

[Zheng et al., 2009] Zheng, S., Dmitriev, P., and Giles, C. L. (2009). Graph based crawler seed selection. In *Proceedings of the 18th international conference on World Wide Web*, WWW'09, New York, USA.

[Zhu et al., 2008] Zhu, J. J. H., Meng, T., Xie, Z., Li, G., and Li, X. (2008). A teapot graph and its hierarchical structure of the Chinese web. In *Proceedings of the 17th International Conference on World Wide Web*, WWW'08, pages 1133–1134, New York, NY, USA. ACM.

[Zliobaite et al., 2011] Zliobaite, I., Bifet, A., Pfahringer, B., and Holmes, G. (2011). Active learning with evolving streaming data. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, ECML/PKDD'11, Berlin, Heidelberg. Springer Berlin Heidelberg.