QUALITY OF THE XML WEB

DRAFT

Steven Grijzenhout, Churchill-laan 25-1, 1078 DC Amsterdam, The Netherlands, 5691982,
sgrijzen@science.uva.nl

Master thesis
MSc Information Science, Business Information Systems
Faculty of Science, University of Amsterdam, The Netherlands
Amsterdam Business School, University of Amsterdam, The Netherlands

Supervisor: dr. Maarten Marx

July 2010

Abstract

XML collections from the web have been previously studied statistically, but no detailed information about the quality of the XML documents on the web is available to date. We have tried to address this shortcoming in this study. An XML collection was gathered from the web and analyzed its quality. The quality of the documents in the collection is surprisingly good; 85.4% of documents is well-formed, 99.47% of all specified encodings is correct, 8.4% of documents that reference a DTD validates with it and 57.9% of documents that reference an XML Schema validates. Furthermore, certain HTTP headers and domain name extension tend to predict valid XML documents. The research will give directions to future research on XML, XML Schema languages and applications using XML.

Keywords: XML, XML Web, DTD, XML Schema, well-formedness, validity.

Contents

Abstract 1
Contents 2
1. Introduction 4
2. Problem definition 5
2.1 Goal 5
2.2 Main research question 5
2.3 Outline 6
3. Background 6
3.1 Studies on XML collections 6
3.2 Studies on HTML web quality 7
3.3 Studies on XML schema languages 8
4. Data 9
4.1 Desired Data 9
4.2 Collecting the Data 10
4.2.1 Crawling a list of URLs 10
4.2.2 Downloading the content of each URL 15
4.2.3 Organizing the collection 15
4.2.4 Determining duplicates 17
4.3 Description of Data 21
4.3.1 General statistics 21
4.3.2 Duplicate statistics 22
4.3.3 Description of URL Data 24
4.3.3.1 Site Distribution 24
4.3.3.2 Document Distribution 26
4.3.3.3 Document Size Distribution 26
4.3.4 Description of Header Data 27
4.3.4.1 General header information 27
4.3.4.2 File size 28
4.3.4.3 Encoding 29
4.3.5 Description of Content Data 31
4.3.5.1 Schema References 31
5. Quality Analysis of Data 33
5.1 XML Well-formedness 33
5.1.1 Fatal errors 36
5.1.2 Recoverable Errors 42
5.1.3 Warnings 43
5.2 Encoding 43
5.3 XML Validity 46
5.3.1 DTDs 47
5.3.1.1 Validating XML with DTDs 48
5.3.2 XSDs 53
5.3.2.1 Validating XML with XSDs 54
5.3.3 Predicting valid XML documents 56
6. Conclusions 61
7. Discussion 63
Availability of data 63
References 63
Appendix 1 – Encodings Found 73

Appendix 2 – Encoding Detection methods.....74
Appendix 3 – Xmllint error domains 75
Appendix 4 – Overview of libxml2 error categories.....75

1. INTRODUCTION

In this study we will look at the prospects of XML technology and XML tools on data found on the World Wide Web. XML, said to be *the* next language on the web, is used in almost all spheres of human activities, and will be used by an ever-increasing number of devices and applications (Toman & Mlynková, 2006; Sundaresan & Moussa, 2001; Lee & Chu, 2000).

For XML to be useful, it is important that the XML documents adhere to certain standards. The well-formedness of XML documents is one of the most important basic characteristics of an XML document because it cannot be used otherwise. Furthermore, the allowed structure of XML documents can be described using XML schema languages, such as DTD and XML Schema (Mlynkova, Toman, & Pokorny, 2006), and XML document needs to stick to the structure as written in its referenced schema. Lastly, more general quality characteristics are important for the delivery and usability of the XML documents found on the web. These are, for example, using the correct encoding, and supplying the right header information. Well-formed and valid XML documents have several important benefits and are often required by applications to function correctly. Initiatives as RDF (W3C, 2004) rely on strict XML documents. Another example is RSS feeds that influences the way people stay up to date and plays an important role in many people's every day life. From an economic perspective, the quality of XML documents on the web will influence economic costs, where high quality documents will result in lower costs as it reduces operating and supporting costs.

The fact remains that documents found on the web, however, are often of *poor* quality. For example, XML's predecessor HTML is still the document format of choice on the internet and several studies have shows that the vast majority of HTML documents (around 95%) on the web did not comply with the standards set by the World Wide Web Consortium (Ofuonye, Beatty, Dick, & Miller, 2010; Chen, Hong, & Shen, 2005; Pollach, Pinterits, & Treiblmaier, 2006).

While the quality of XML documents become increasingly important, the part of the web that consists of XML documents, the XML web is not studied on its quality. It is studied statistically (Mignet, Barbosa, & Veltri, 2003; Mlynkova, Toman, & Pokorny, 2006), but no detailed information about its quality is available to date. In this study, we will assess this shortcoming and take a closer look at the quality of an XML collection that was harvested from the XML web. The study includes the assessment of the well-formedness of the documents, as well as their validity. The results include most common errors and validation problems. Having this detailed information can provide direct benefits to technologies using XML and those using the XML web in particular.

2. PROBLEM DEFINITION

A vast share of today's applications rely on data sources available on the internet. Such applications combine and re-use the data and provide new services that were not previously possible. In popular terms, these applications are called *mashups*. There are countless examples; Google Maps combines information based on location, and offers an accessible way to browse based on location. Personal pages as iGoogle offer an easy overview of your day, and RSS feed readers provide an easy way to stay up to date about several different news sources.

All these mashups rely on data exchange: they combine information from different sources. For this to work properly, the mashups needs to know in what form they can expect the data and what the structure will look like. This is what XML provides: a way to facilitate data exchange. Mashups can combine data from API's (Application Programming Interface) that are set up to interact with the data source. When an API is not available, one can harvest data from a data source itself.

Needless to say, the quality of the XML that is used needs to be of high quality in order to be used by mashups. Due to the nature of the XML specification, an XML document that is not well-formed is useless to a mashup and can even cause applications to stop functioning properly because it cannot use this data and will force the application to stop processing. In addition, it is important that the mashups know what to expect in order to combine the right information in the data. Therefore an XML document needs to be described by an XML schema language, and it needs to adhere to this schema. For example, a RSS reader needs to know where to find the publication date of a news item for it to be able to correctly order the news item from different sources.

2.1 Goal

The goal of this research is to gain insight into the quality of XML documents found on the web today. We will be most interested in documents of poor quality. As is the case with the HTML documents on the web, there is a high probability that a vast amount of XML documents on the web will be of poor quality. We want to gain insight into the most common errors in well-formedness and validity of the documents. Subsequently, we want to gain insight in possible correlations and regressions between document characteristics and errors. This way we can, for example, investigate whether documents that are using a specific encoding have more chance to contain errors than others.

2.2 Main research question

The main research question of this study is:

> What is the current quality of XML documents on the web?

The term quality in this question consists of the correctness of properties of the documents, the well-formedness and the validity of the documents.

2.3 Outline

The remainder of this document is used to describe the approach used to tackle the main research question. In section 3 the background of literature is summarized in a small fashion. In section 4 the data that is used is described; a detailed description of the harvesting and organizing of the collection is given, and it will conclude with an extensive description of properties and characteristics the final data collection used. Thereafter, in section 5 the analysis of the quality of the data collection will be discussed; that section is the core of this paper. Finally, we will draw conclusions in section 6 and provide a discussion in 7 where we will also give suggestions for further research.

3. BACKGROUND

Due to its powerful characteristics and its flexibility, XML has become one of the standards in data management and data exchange. Consequently, a large number of studies on XML has been conducted. There are three main themes identifiable in the literature, which will be discussed accordingly. First, studies on the XML collections will be discussed. Secondly, studies on the quality of the HTML web will be discussed. Lastly, we will discuss studies on XML schema languages.

3.1 Studies on XML collections

Studies on XML document collections mainly differ in sample data (Toman & Mlynková, 2006). A study has been done on 200,000 publicly available XML documents from the Xyleme repository (Mignet, Barbosa, & Veltri, 2003). This collection consists of only well-formed XML documents. Another study used a number of XML collections, consisting of 16,534 documents and accounting for a total size of 20 Gigabytes (Mlynkova, Toman, & Pokorny, 2006). The collections include well-known docbook samples, XML bibles, RDF samples and IMDb collections. Lastly, a third study used 601 XHTML web pages, 3 DocBook XML documents, and documents from the XML Data repository[1] project (Kosek, Kratky, & Snasel, 2003).

[1] XML Data repository: http://www.cs.washington.edu/research/xmldatasets/

Macro-level analysis shows that XML documents are found in all geographic regions and all across all major internet domains. 53% of all documents, accounting for 76% of the total file size, can be found at '.com' and '.net' internet domains (Mignet, Barbosa, & Veltri, 2003).

Only 48% of the XML documents references a DTD, and 0,09% an XML Schema (Mignet, Barbosa, & Veltri, 2003).

Most XML documents are small: around 4 Kilobytes. Also, the volume of markup in relation to the actual content of the documents is surprisingly high. Lastly, 99% of the documents had less than 8 levels of element nesting, and 15% appears to have recursive content. This all seems to indicate that most XML documents are not complex (Mignet, Barbosa, & Veltri, 2003; Kosek, Kratky, & Snasel, 2003).

3.2 Studies on HTML web quality

Surveys on the quality of HTML documents on the web exist (Ofuonye, Beatty, Dick, & Miller, 2010; Chen, Hong, & Shen, 2005; Beckett, 1997; Pollach, Pinterits, & Treiblmaier, 2006). Although XML's predecessor HTML differs greatly in applicability, these studies are relevant mostly because of their approach.

The sample data across the studies differ between 226 web sites from environmental issues (Pollach, Pinterits, & Treiblmaier, 2006), 13,312 websites under the 'co.uk' domain (Beckett, 1997), samples that combined websites from search engines and Alexa.com's top web sites (Chen & Shen, 2006), and homepages of the Alexa.com's top 100,000 web sites (Ofuonye, Beatty, Dick, & Miller, 2010).

The studies use different methods to assess the quality of HTML documents: WebXACT (Pollach, Pinterits, & Treiblmaier, 2006), NSGMLS parser (Beckett, 1997) and the W3C HTML Validator (Chen, Hong, & Shen, 2005; Ofuonye, Beatty, Dick, & Miller, 2010).

The differences in sample collections and quality measures does not seem to make a big difference in results as all indicate a poor quality of the HTML documents: a mere 6.5% (Beckett, 1997), 5% (Chen & Shen, 2006), 4% (Pollach, Pinterits, & Treiblmaier, 2006) and 3% (Ofuonye, Beatty, Dick, & Miller, 2010) of HTML documents complied with HTML standards.

3.3 Studies on XML schema languages

XML schema languages can describe the structure of XML data. And indeed, schemas are almost inseparable from XML. They allow automization and optimization of search, integration and processing of XML data (Bex, Neven, Schwentick, & Tuyls, 2006).

In actuality, there are three main schema languages and one language for specifying dependencies in use. These are DTD[2], XML Schema[3], Relax NG[4] and Schematron[5]. Schematron is the language used for expressing dependencies in the form of implications between tree patterns. As it is rarely used and we will not discuss it further. The three other schema languages are all W3C recommendations. XML Schema and Relax NG documents are themselves written in XML. DTDs have their own syntax. No research yet exists on the actual use of Relax NG schemas in documents on the web.

Several different approaches to studying XML schemes can be found. Firstly, XML schemes are studied in relation to XML collections. As we have seen above, only a percentage of documents reference a schema. In the Xyleme sample 48% of the documents references a DTD, and 0.09% an XML Schema (Mignet, Barbosa, & Veltri, 2003). In the semi-automatic collection by Mlynkova, Toman, & Pokorny (2006), however, only 7.4% does not reference a schema; this might be due to the collection process. Furthermore, results show that the XML documents are simple and specific in comparison to their XML schema; the schema is usually too general.

As is the case with HTML files, the syntax of most DTD files is incorrect (Sahuguet, 2001; Choi, 2002). This is generally also the case for other schema languages as XML schema (Bex, Neven, & Bussche, DTDs versus XML Schema: A Practical Study, 2004).

Secondly, the properties of XML schemes are studied. Most of this work has focused on DTDs. DTDs can be categorized in many different ways, among others by its intended use (Choi, 2002). Between these categories, the properties of DTDs tend to differ. Overall, DTDs differ greatly in size and forms. However, DTDs are generally simple (Choi, 2002; Klettke, Schneider, & Heuer, 2002). Many features of DTDs are not used or misused; this indicates that the features are not properly understood. Also, there are many ways to do things in DTDs, and people use hacks to cope with DTD shortcomings (Sahuguet, 2001).

---

[2] http://www.w3.org/TR/REC-xml/#dt-doctype
[3] http://www.w3.org/XML/Schema.html
[4] http://www.relaxng.org/
[5] http://www.schematron.com/

Thirdly, work has been done in developing metrics to measure the properties of DTDs (Klettke, Schneider, & Heuer, 2002) and XML Schemas (McDowell, Schmidt, & Yue, 2004). These metrics might be interesting to use in future versions of quality analysis.

Lastly, research is done in comparing the use of the different XML schema languages. The three languages are incomparable in expressive power and their effect when validating. For instance, validating a document with a DTD changes the document: default values are added. DTDs have no means to restrict data values to data types like string or integer. Theoretical work on the expressive power of schema languages abstract many features of the concrete languages and compare their core logical part. Then we see that DTD is less expressive than XML Schema, which is less expressive than Relax NG, which is equally expressive (on XML trees) as Monadic Second Order Logic (MSO) (Bex, Martens, Neven, & Schwentick, 2005; Martens, Neven, Schwentick, & Bex, Expressiveness and complexity of XML Schema, 2006). While XSDs allow expressions that cannot be expressed in DTD syntax, these extras are rarely used in practice (Bex, Neven, & Bussche, DTDs versus XML Schema: A Practical Study, 2004; Martens, Neven, & Schwentick, Which XML schemas admit 1-pass preorder typing?, 2005).

4. DATA

4.1 Desired Data

The population of the data in this study is the *XML Web*. The definition for the XML Web used here will be: *the subset of the web made of XML documents only* (Barbosa, Mignet, & Veltri, 2005, p. 2). The population data consists of all kinds of XML documents. RSS, Atom, XSL stylesheets, XSD data and XHTML are all written in XML, and are therefore part of the population the XML Web.

The actual amount of files in the XML web is unknown. Obtaining an estimate of its size is intrinsically difficult (Abiteboul & Vianu, 1997 in Barbosa, Mignet, & Veltri, 2005). The size of the population is, however, irrelevant in calculating a representative sample size. Unfortunately, collecting XML documents from the web is often *not* a simple random sample. Because of this, it is not possible to calculate a required sample size, as these traditional statistical methods require a simple random sample. In the discussion section we will have a closer look into this issue.

We decided to harvest a large collection of XML documents from the web. The objectives of this research are to assess the quality of the XML Web, and a large collection will maximize the probability that errors are included in the collection.

A last issue we want to address is the *Hidden Web* (Raghavan & Garcia-Molina, 2001). The Hidden Web is the part of the web that is not publicly indexable because it is not reachable by purely following hyperlinks. Examples of obstructions can be forms that have to be submitted first, or authorization that is required. The Hidden Web is particularly important as it contains high-quality information in databases, for example from patent bureaus (Raghavan & Garcia-Molina, 2001). Our collection process does not take in account the Hidden Web. As a consequence, our collections will not contain any data from the Hidden Web. We note that crawling the Hidden Web requires special tools and human guidance (Barbosa, Mignet, & Veltri, 2005; Raghavan & Garcia-Molina, 2001).

4.2 Collecting the Data

The data collection process consisted of the following steps:

1. Crawling a list of URLs of XML documents from Yahoo and Google,
2. Downloading the content of each URL,
3. Organizing the collection,
4. Determining duplicates.

4.2.1 Crawling a list of URLs

The first step consisted of creating a list of URLs with XML documents. This list was created using a modified version of the crawler by Bex, Neven, & Bussche (2004). The crawler executes several queries on the XML-filetype, and saves the URLs from the result pages to a Java treeset. The crawler subsequently saves the treeset in a local file.

Only Yahoo and Google were used as they provided a function to limit the search to a specific filetype. Bing was considered, but only provides this function for specific filetype as PowerPoint and Excel[6].

The queries that were used consisted of 12 different types. We will first discuss queries on Yahoo. Four different queries were implemented for Yahoo.

```
QUERY TYPE 1:
originurlextension:{extension}
```

[6] Bing, Version 2.0 API. WebRequest.FileType Property. http://msdn.microsoft.com/en-us/library/dd250876.aspx. The following extension were supported at the time of writing: DOC, DWF, FEED, HTM, HTML, PDF, PTT, PS, RTF, TEXT, TXT and XLS.

Where {extension} is the filetype extension, in this case '.xml'. It takes approximately 50 seconds to execute the query and save all result pages. Every query returns around 10 result pages with 100 results, resulting in almost 1000 result URLs.

QUERY TYPE 2:
originurlextension:{extension} {a-z}

Where {a-z} are the lowercase characters ranging from a to z. The query is executed separately for every lowercase character in the alphabet, and for every query all URLs from all result pages are saved. Figure 1 shows the return of unique URLs on query type 2. Each query returned between 682 and 986 unique URLs. One would expect that the amount of unique URLs returned on a letter at the end of alphabet would be less than one at the beginning, because an overlap in results would appear, and only new URLs are saved. This is not necessarily the case; the letter x for example, returned 826 unique URLs, while the letter r added only added 682 unique URLs to the list. The trend line however, shows that clearly the amount of unique URLs added to the set decreases over time.

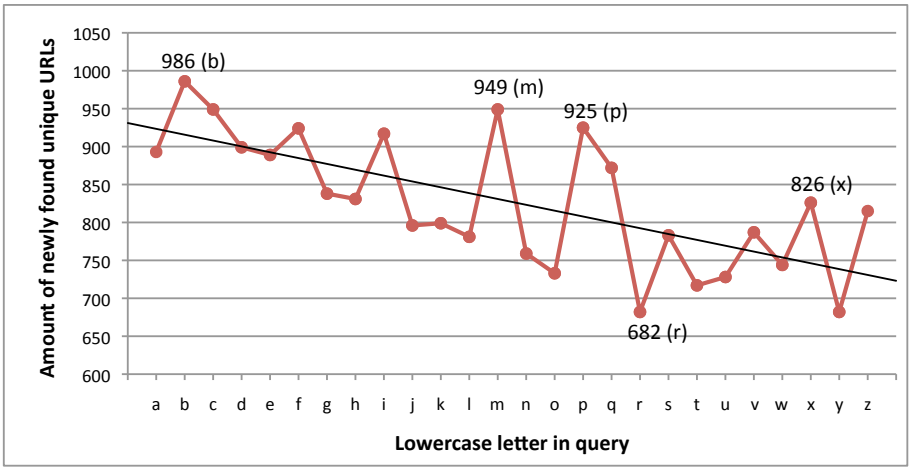

Figure 1. Amounts of Unique URL Results with Query Type 2 on XML.

QUERY TYPE 3:
originurlextension:{extension} &region={yahooRegions}

Where {yahooRegions} are the country regions supported by Yahoo[7]. The query is executed separately for every country region, and for every query all URLs from all result pages are saved. The

[7] All supported country regions can be found at: http://developer.yahoo.com/search/regions.html

visualization of the unique URL results of every query in Figure 2 shows that most of the countries add between 900 and 1000 unique URLs to the set.

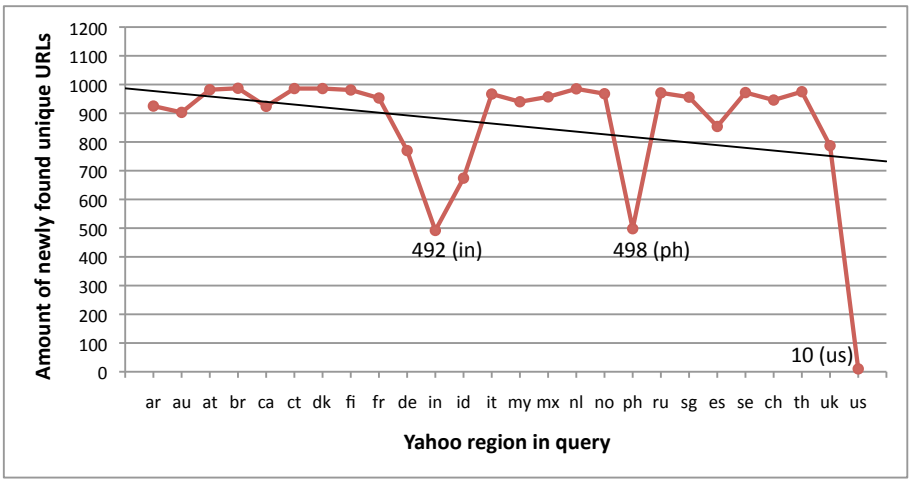

Figure 2. Amounts of Unique URL Results with Query Type 3 on XML.

Exceptions are India, the Philippines and the USA, adding respectively 492, 498 and 10 new URLs to the list. The low result of the USA can be explained by the fact that Query type 1 and Query type 2 were executed on google.com, which would provide results in favor of the USA. Subsequently, the overlap with the results from the USA at Query type 3 would be higher than other countries.

QUERY TYPE 4:

originurlextension:{extension} {a-z} &region={yahooRegions}

This query is executed separately for every unique combination of a country region with a character.

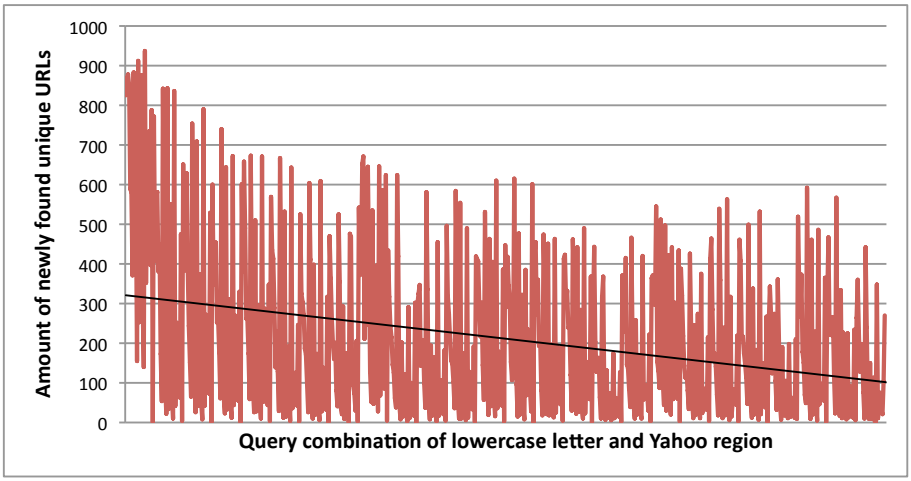

Figure 3. Amounts of Unique URL Results with Query Type 4 on XML.

The result of new URLs from Query type 4 is visualized in Figure 3. The trend line clearly shows that overlap occurs which pushes the amount of new URLs at the end of the Query type to an average of fewer than 100 new URLs per query.

The Google queries consisted of the remaining eight types.

QUERY TYPE 5:
filetype:{extension}

Where {extension} is the filetype extension, in this case '.xml'. This query is equivalent to the Yahoo Query type 1. Unfortunately, this query resulted in a total of 0 (!) new URLs added to the list of XML files. The results from this query probably completely overlapped previous results from the Yahoo searches.

QUERY TYPE 6:
filetype:{extension} {a-z}

Where {a-z} are the lowercase characters ranging from a to z. This query is equivalent to the Yahoo query type 2. The whole query type on XML resulted in only 14 new URLs added to the list. Each letter accounted for a maximum of 1 new URL. The visualization can be viewed in Figure 4.

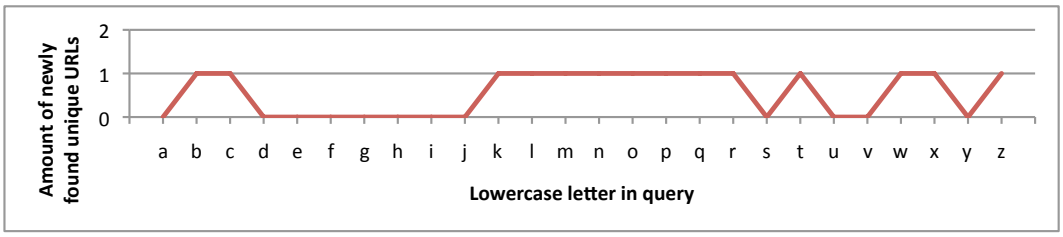

Figure 4. Amounts of Unique URL Results with Query Type 6 on XML.

QUERY TYPE 7:
filetype:{extension} daterange:{date range}

Where {date range} is a date range in the form of date1-date2 ranging from 0 to 2454766, with interval of 50. Again, the results from this query probably completely overlapped by previous results from the Yahoo searches, and this query type results in 0 new URLs.

QUERY TYPE 8:
filetype:{extension} {a-z} &gl={googleCountryCodes}

Where {googleCountryCodes} are the country regions supported by Google[8]. Note that the gl-parameter boosts queries in the specified country, but does not limit results to the country (as the cr-parameter does). For future queries experiments can take place with both parameters. Again, 0 new URLs were found. The remaining Google Query Types 9 to 12 also resulted in 0 new URLs.

QUERY TYPE 9:
filetype:{extension} {a-z} daterange:{date range}

QUERY TYPE 10:
filetype:{extension} daterange:{date range} &gl={googleCountryCodes}

QUERY TYPE 11:
filetype:{extension} {a-z} &gl={googleCountryCodes}

QUERY TYPE 12:
filetype:{extension} {a-z} daterange:{date range} &gl={googleCountryCodes}

Yahoo and Google both have a limit on the amount of requests accepted during a certain time span. To overcome this, a timer has been built in that waits between 1 and 60 seconds between each request. This results in an average waiting time of 30 seconds. In the future, the waiting time of the crawler could be calculated with the amount of request that are allowed by the search engines to optimize the queries, as execution time increases substantially with each second of extra waiting time between queries.

The crawling of the URL list resulted in the lists described in Table 1.

[8] Supported country codes by Google can be found at http://www.google.nl/cse/docs/resultsxml.html#countryCodes

Table 1

Statistics of URL List and Downloading

Filetype	Unique URLs in List	Files Downloaded	Loss Percentage	Last File Downloaded
XML	188,332	180,640	4.08%	2010-07-17

4.2.2 Downloading the content of each URL

The second step in the data collection process was the actual downloading of the contents of the URLs that were collected in the first step.

The program wget[9] was used to download the contents of the URL files. The following command was used:

```
wget $url --restrict-file-names=unix --timeout=3 --quiet --save-headers --output-document=$document
```

A script was written that looped over each URL that was collected in the previous step. It downloads the file using the above wget command, and saves it with an id as local filename on the local file system.

To speed up the process, the list of URLs was split up into several smaller lists, which where downloaded in parallel. The machine that was used ran on Fedora Linux, had 8 cores, and 1TB disk space. Also important is the fact that the internet connection of the University of Amsterdam was used, providing huge bandwidth.

Table 1 provides an overview of the statistics of the downloading. It provides the amount of files that were actually downloaded. Due to several reasons, not every URL in the list could be downloaded. These reasons include, among others, 404 errors and read errors. The percentage of loss of downloaded files as opposed to the original URL list is also noted in the table. Finally, the date of download of the last file is noted to provide a measure of out-datement.

4.2.3 Organizing the collection

The resulting collection was organized in a relational MySQL database. The schema consists of nine main relations, which are described, in Figure 5.

[9] http://www.gnu.org/software/wget/

file	**header**	**duplicate_def5**
id (*int, key*)	id (*int, key*)	fileid (*int, foreign*)
URL (*text*)	author (*text*)	duplicate (*int, foreign*)
basedomain (*text*)	cache-control (*text*)	
domainextension (*text*)	...	
calculatedfilesize (*int*)	..	
filenameextension (*text*)	.	

encoding	**schema_dtd**	**schema_xsd**
id (*int, key*)	id (*int, key*)	id (*int, key*)
headerencoding (*enum*)	schemaid (*int, foreign key*)	schemaid (*int, foreign key*)
headerencodingchecked (*enum*)	fileid (*int, foreign key*)	fileid (*int, foreign key*)
pseudoattrencoding (enum)	reconstructedurl (*text*)	reconstructedurl (*text*)
pseudoattrencodingchecked (*enum*)	compiles (*enum*)	compiles (*enum*)
metatagencoding (*enum*)		
metatagencodingchecked (*enum*)		

xmllint_wellformedness_errors	**xmllint_validity_errors_dtd**	**xmllint_validity_errors_xsd**
id (*int, key*)	id (*int, key*)	id (*int, key*)
fileid (*int, foreign*)	schemaid (*int, foreign*)	schemaid (*int, foreign*)
linenumber (*int*)	fileid (*int, foreign*)	fileid (*int, foreign*)
errordomain (*enum*)	linenumber (*int*)	linenumber (*int*)
errorlevel (*enum*)	errordomain (*enum*)	errordomain (*enum*)
errortype (*enum*)	errorlevel (*enum*)	errorlevel (*enum*)
specificinformation (*text*)	errortype (*enum*)	errortype (*enum*)
entityline (*text*)	specificinformation (*text*)	specificinformation (*text*)
element (*text*)	entityline (*text*)	entityline (*text*)
	element (*text*)	element (*text*)

Figure 5. Structure of the database; the nine main relations with their attributes.

The actual files are saved on file system with the appropriate id as its filename. For example, the id of the file-relation corresponds with the file <id>.xml on file system. The same is the case for schema_dtd that corresponds with <id>.dtd on file system, and schema_xsd, where id corresponds with <id>.xsd on file system.

Some headers contained duplicate property keys. For example, the property set-cookie might be given on the same document multiple times with different values. Because the properties that had duplicates were deemed of secondary importance to the analysis, it was decided to concatenate the

values instead of creating a separate relation. This way, it was possible to keep the queries on the database relatively simple. The values of duplicates were separated by a ":", and inserted into the same data field. Because this is a reserved character, the original distribution of headers can be reconstructed when appropriate.

Furthermore, a distinction is made between a header that is being send by the server but has no value, (which in the database will be a string of length 0), and a header that is not send at all (which in the database will be of value *NULL*).

During the creation of the database, several errors in the headers have been found, which had to be resolved before the data could be inserted into the database. The first is the fact that two documents had header-keys that were empty, and each of these header values was empty also. Such a header might look like the following:

```
:
```

Because a column-name of length 0 is not allowed, the column is named EMPTY. Secondly, while parsing the header keys to the database, white space at the beginning and end of the keys have been discarded, because MySQL does not allow this in a name of a column. Thirdly, special characters were normalized to ASCII format. An overview of alterations on header keys that have taken place before insertion in the database can be found in Table 2.

Table 2

Overview of alterations of header keys before insertion in database.

Property as Found in Header	Property as Inserted in Database	Reason for Change
``	`EMPTY`	Empty string not allowed as a column name
` key`	`key`	White space not allowed at beginning of a column name
`key `	`key`	White space not allowed at end of a column name
`codificación`	`codificacion`	Special characters are normalized to ASCII

4.2.4 Determining duplicates

The next step consisted of determining duplicates. Different definitions of duplicates exist; we must determine which one to use. In this section we will first define the separate entities that a document consists of. Next, we will discuss the possible definitions of equality of each individual entity.

Following that, we will define the definitions of equality of whole documents. Finally, we will present the one we will use in this research.

We define that a document consists of three entities: URL, header and content. This definition is summarized in Definition 1.

> DEFINITION 1:
> Document D = <URL, header, content>

The *URL* is the location from which the document is retrieved. The *header* consists of the HTTP response headers that are being received from the server that sends the document. The document's *content* consists of the actual content of the document. This can be XML for an XML document, or DTD for a DTD document, etcetera. A visual representation of Definition 1 can be found in Figure 6 to clarify.

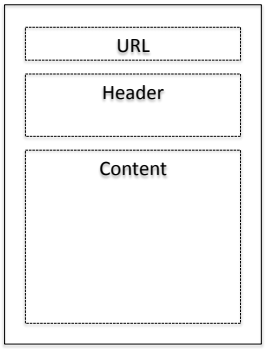

Figure 6. Visual Representation of a Document as defined in Definition 1.

We will first look at an individual level at the equality of the elements: URL, header and content respectively.

Duplicate *URLs* are easily identified because an URL is unique by design. However, the document from the same URL may still change over time. That is why the equality of a document cannot be solely defined by the equality of an URL.

The values of header properties are compared to determine whether two *headers* are identical. The 'date'-property is excluded from this comparison, as this indicates when the document was retrieved from the server instead of when the content was changed. The following definition of header equality will be used:

> DEFINITION header equality:
> X.header = Y.header iff (X.header – X.header.date) = (Y.header – Y.header.date)

In regard to the equality of the content of a document, several options exist. The first is a byte-for-byte comparison between two files. A difference in one white space character, however, will already make that two document are not deemed equal anymore. We have the choice to disregard all white space or to preserve white space in the comparison. This first option is used in (Mlynkova, Toman, & Pokorny, 2006), and the second in (Barbosa, Mignet, & Veltri, 2005).

We are aware that several different definitions of white space exist. The W3C XML 1.0 specification[10] defines that white space *S* consists of one or more space characters, carriage returns, line feeds, or tabs:

> DEFINITION W3C White Space:
> S ::= (#x20 | #x9 | #xD | #xA) +

Next to this, the W3C XML 1.0 specification makes a distinction between significant and insignificant white space. When white space is significant, it should be preserved in the delivery version. This can be desirable in, for example, poetry. When white space is insignificant, it is not intended for preservation in the delivery version. Whether white space should be preserved or not can be controlled by using a DTD or XML Schema. On default, white space should be preserved; this is also the case when no DTD or XSD is specified. Because we presume that XML data that is publicly available on the web does not strictly adhere to the definition of significant and insignificant whitespace, we chose to disregard all whitespace in the comparison. This resulted in the following definition of content equality:

> DEFINITION 5 content equality:
> X.content = Y.content iff (X.content– X.white space) = (Y.content – Y.white space)
> WHERE white space = significant and insignificant white space as defined in Definition W3C White Space.

We acknowledge that, next to white space, several other problems with the above definition exist. For example, single quotes or double quotes are both allowed to define attributes in XML. This means that the following expression should be true, but using our definition of content quality it is not:

> <element attribute='value' /> = <element attribute="value" />

[10] Extensible Markup Language (XML) 1.0 (Fifth Edition) <http://www.w3.org/TR/REC-xml/>. Section 2.3 Common Syntactic Constructs.

The W3C definition on XML copes with these problems by building a model of the XML documents and subsequently comparing these models. While documents might be deemed equal using the W3C document, the documents cannot be discarded, as using single our double quotes might still influence properties of document, for example the file size. Similar W3C equality definitions exist for DTD, XSD, RNG and RNC documents.

We decided to use the content equality Definition 5 because of the following reasons. Firstly, the definition is used in other similar research (Barbosa, Mignet, & Veltri, 2005; Mlynkova, Toman, & Pokorny, 2006). Secondly, the calculations of equality using the W3C definitions are computationally expensive. Thirdly, the W3C definitions are only usable for well-formed and valid XML. We presume that XML data that is publicly available on the web does not strictly adhere to those definitions, and would loose all interesting data for the objective of this research.

We have determined the definitions of equality of each individual entity in a document. Several combinations of entities within a document do, however, still have effect on the equality of a document as a whole. For example, the content of a document may be equal using our definition, but the header of both documents is not. This will mean that the document is not equal despite the fact that the content is the same.

We will walk through several combinations of each entity and finally choose a definition of document equality.

The first definition of document equality regards two documents as equal when the content of the documents is identical. The Xyleme XML dataset as used in Barbosa et al. (2005) is an example, and the DTD and XSD collections used by Bex et al. (2004) another. This definition can be defined as:

DEFINITION 2: Xd = Yd iff X.content = Y.content

Secondly, we can define two documents as equal when the content and the header are identical. This is a relevant definition when one considers automatically generated XML documents as identical, but still wants to investigate different systems on which it runs (through, for example, the ‘server’-property in the header). This definition can be notated as:

DEFINITION 3: X = Y iff X.header = Y.header AND X.content = Y.content

Note that we use header equality as defined above, so the ‘date’-property is not taken in account.

Thirdly, we can define two documents as equal when the content, the header and the URL are equal. Because a URL's content and header information can change over time, and thus the document provided by a URL, the definition also includes header and content equality. This can be summarized as the following definition:

DEFINITION 4: X = Y iff X.url = Y.url AND X.header = Y.header AND X.content = Y.content

The definition that will be used in the description and analysis of the collection depends on the suitability to the specific analysis. For example, if we make a scatter plot of file sizes, we will use definition 2, as we do not want automatically generated documents to give a huge peak. In contrast, we will use definition 4 to do analysis of domain name regions and the validity of the documents.

Duplicates were *not* removed from the dataset, but rather a relation of duplicate content using definition 5 was inserted into the database (see relation Duplicate_def in Figure 5). This allows us to analyze the dataset given all the definitions, while still being able to use the dataset in future research based on other definitions of equality by simple creating a new relation representing another definition of equality.

4.3 Description of Data

This section will describe the data collection. This section is divided into four main paragraphs. First some general statistics about the collection will be presented. This includes the amount of files, the file size of the entire collection, and a comparison with XML data collections of similar studies. Secondly, more specific details about the URLs of the documents will be discussed. This includes descriptions of the distribution of the collection across top-level domains and world regions. Thirdly, descriptions of the headers of the documents will be discussed. This includes the different headers we came across, as well as a more specific description of the most important headers. Lastly, the content of the documents will be discussed.

4.3.1 General statistics

The XML collection contains 180,640 XML files. Table 3 shows that this is 5.1% smaller than the collection used by Barbosa et al. (2005), but 992.3% larger than the collection used by Mlynkova et al. (2006). The total file size of the collection is 40 Gigabytes in uncompressed form. Compressed in tar.gz format, the collection accounts for 4.1 Gigabytes of disk space. The largest file in the collection is 683.7 Megabytes, and the smallest is 1 byte. The average file size is approximately 223 Kilobytes. An overview of comparisons can be viewed in Table 3.

Table 3

Comparison of XML collections

	Grijzenhout	Barbosa	Mlynkova
Amount of files	180,640	190,417	16,538
Source	Selected from queries by Google and Yahoo	Randomly selected from Xyleme public repository (approx 500,000 files)	Semi-automatically from manually selected XML collections
Total size uncompressed	40 Gigabytes	843 Megabytes	20,756 Megabytes
Total size compressed (.tar.gz)	4.1 Gigabytes	151.4 Megabytes	--[a]
Amount of websites	96,650	19,254	133 Collections
Amount of duplicates	2430	26,989	--[a]
Type of duplicate detection	Hashing algorithm disregarding white spaces	Fingerprinting techniques	Simple hashing algorithm disregarding white spaces
Preprocessing	None	Collected from Xyleme database, which consists of well-formed XML files only.	Manually fixed most errors on XML, DTD and XSD files (including well-formedness, encoding & wrong usage of namespaces) Computer generated and random-content XML files were eliminated.

[a] The value is unknown.

4.3.2 Duplicate statistics

Using the definition of document equality number 5, a total of 2430 documents are a duplicate of another document (one of the documents having duplicates is defined as unique, and is not added to the sum). This is only 0.0135% of the collection, which is significantly lower than documents in comparable studies (although we are not exactly sure how they counted duplicates). The total amount of documents that has a duplicate is 1296. This means there is chance of 0.007% that a document has a duplicate. The highest number of duplicates of one file is 119 duplicates, and the lowest 1 duplicate. As can be seen in Figure 7, only a small proportion of files have a lot of duplicates and the tail is approaching 0 fast.

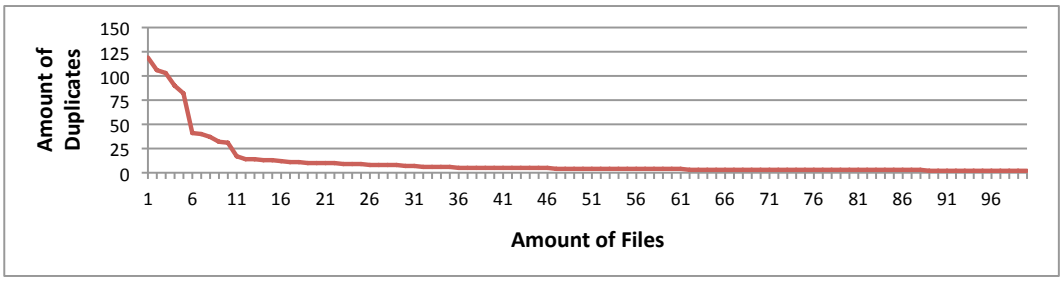

Figure 7. Distribution of Duplicates.

Figure 8 shows that most duplicates are located in '.com'-domains, accounting for 31.7% of all duplicates. The European Union comes second with 31.2%, followed by the category The Rest of the World with 22.9%.

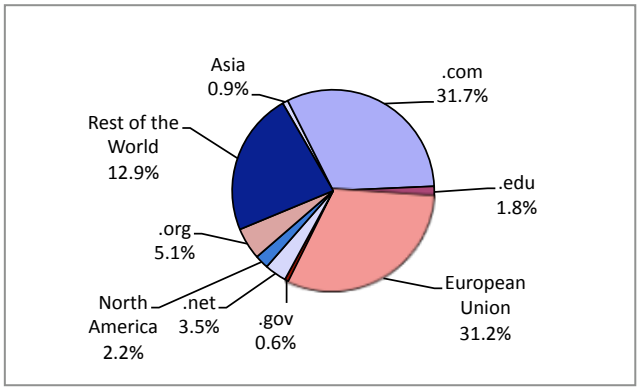

Figure 8. Geographical Distribution of Duplicates.

Regarding base domain, most duplicates are found in the Argentinean domain 'afip.afip.gov.ar'. It is followed by the Italian domain 'cylex.it' with 44 duplicates. Tied in third place are 'familyliving.se' and 'afip.gob.ar' with 31 duplicates. Another long-tail form can be plotted with base domain, as can be seen in Figure 9.

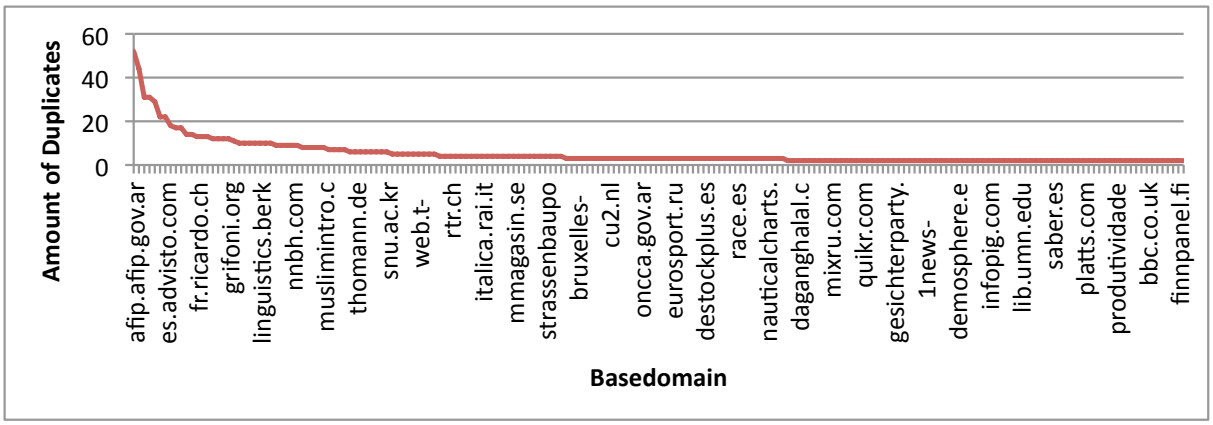

Figure 9. Distribution of Duplicates per Base domain.

4.3.3 Description of URL Data

The URLs in the collection allow us to describe the distribution of XML documents on the web. The regions from which the XML web is hosted and served can be explored and, up to a point, the underlying institutional goals of the XML can be described. For example XML documents hosted on educational domains or commercial domains.

4.3.3.1 Site Distribution

The URL of a document contains the site from which it was retrieved. In this case, we will define a site as a combination of a base domain and top-level domain. Optional is an extra sub domain that will be placed in front of the base domain. The illustration in Figure 10 demonstrates the structure of a site's identifier.

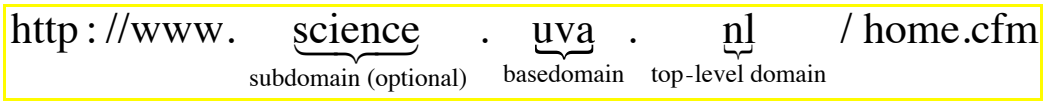

Figure 10: Structure of site domains

To further illustrate how the sites were deducted from the URLs, more examples are given:

http://1script.info/design/rss.xml	→	1script.info
http://20100622032602335.en.hisupplier.com/rss.xml	→	en.hisupplier.com
http://180graus.brasilportais.com.br/rss/augusto-cesar.xml	→	180graus.brasilportais.com.br

There are 96,650 different Web sites in our collection. To gather any meaningful data, we have clustered the results of the Web sites by zones, consisting of generic Internet domains and geographical regions. These zones have been defined by Barbosa et al. (2005), with the only difference that we used the European Union as if June 2010, while Barbosa et al. used the EU as of 2002. An overview is available in Table 4.

Table 4

Definition of Zones

Zone Name	Description	Included Top-level Domains (Amount)
.com	Generic Internet Domain "Commercial"	.com (1)
.edu	Generic Internet Domain "Education"	.edu (1)
.org	Generic Internet Domain "Non profit"	.org (1)
.net	Generic Internet Domain "Network"	.net (1)
.gov + .mil	Generic Internet Domain "Government institutions"	.gov, .mil (2)
Asia	Geographical zone consisting of China, India, Indonesia, Japan, Pakistan, Taiwan, South Korea and Singapore	.cn, .id, .jp, .pk, .tw, .kr, .sg (7)
European Union	Geographical zone consisting of member countries of the European Union as of June 2010	.at, .be, .bg, .cy, .cz, .dk, .ee, .fi, .fr, .de, .gr, .hu, .ie, .it, .lv, .lt, .lu, .mt, .nl, .pl, .pt, .ro, .sk, .es, .se, .uk, .gb (27)
North America	Geographical zone consisting of Canada, the United States and Mexico	.us, .ca, .mx (3)
Rest of the World	All other countries and top level domains	All remaining top-level domains (n - 43) and the ones lacking a top-level domain

Figure 11 shows that 38,197 Web sites (39.5%) in the collection are in the '.com' domain. The EU follows with 25,870 Web sites (26.8%), and the Rest of the World category accounts for 18,753 Web sites (19.4%).

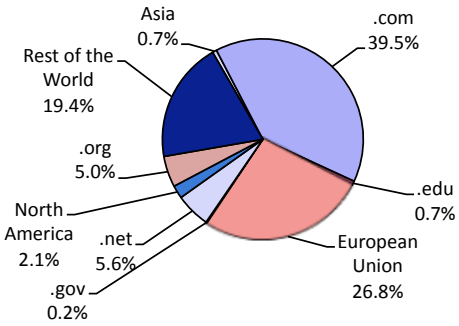

Figure 11. Distribution of Web sites by Zone.

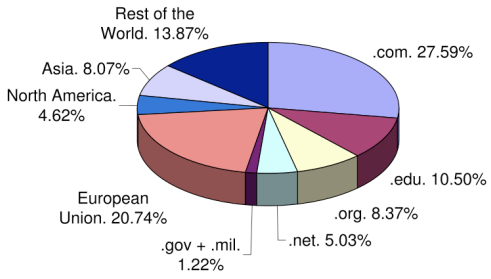

Figure 12. Distribution of Web sites by Zone (Barbosa et al., 2005).

These results are in line with results by Barbosa et al. (see Figure 12), where the top three consists of the exact same zones. However, the results show that in geographical terms North America

(composed of North America, .edu, .gov and .mil) is under represented: it accounts for only 3% in our collection, while it accounts for at least 16% in Barbosa's collection. This might be due to the fact that the harvesting process of our new sample was located in The Netherlands as opposed to North America.

4.3.3.2 Document Distribution

With 180,640 documents and 96,650 Web sites in our collection, there is an average of 1.87 documents per Web site. The site 'gentoo.org' has most documents: 451, followed by 'thomann.de' with 207 documents.

The distribution of documents per zone is represented in Figure 13. The distribution largely mirrors the distribution of Web sites in Figure 11: '.com' leads, followed by the European Union and The rest of the World category. Compared to the collection of Barbosa et al., however, there are only 4.7% of documents in the '.net' zone, while that accounted for more than 20% in Barbosa's collection. This is even more surprising as it accounted for only 5% of the sites in Barbosa's sample. Supposedly a large amount of documents came from the same '.net' sites in Barbosa's sample, while it is more evenly distributed over Web sites in our collection. This can be the result of the fact that we have a lot more Web sites in our collection.

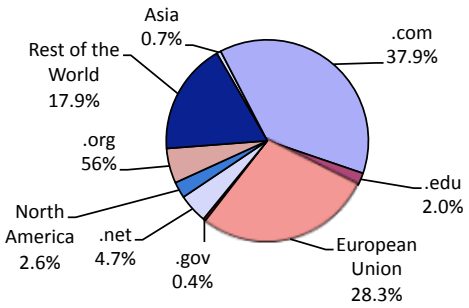

Figure 13. Distribution of Documents by Zone.

4.3.3.3 Document Size Distribution

The document size distribution shows the cumulative amount of Content-Length in bytes per zone. Figure 14 shows results similar to those found previously; '.com' (16.4 Gigabytes, 37.9%), EU (8.7 Gigabytes, 28.3%) and Rest of the World (6.8 Gigabytes, 17.9%) lead the chart.

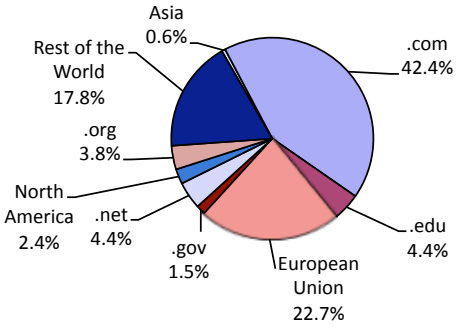

Figure 14: Distribution of Document Size by Zone.

4.3.4 Description of Header Data

4.3.4.1 General header information

In all 51 different kinds of headers have been found. Several alternative spellings of property keys of the headers have been found, these include typos. A list of the alternative spellings of official HTTP 1.1 headers has been summarized in Table 5. Note that case sensitivity has not been taken in account.

Table 5

Most Important Errors in Header Property Key Spelling

Property Key	Alternatives Found	Number of Times Found	Percentage of Use opposed to Valid Property Key (1691)	Percentage of Use opposed to All Properties (1,132,755)
connection	_onnection	13	0.8%	0.0%
	cneonction	78	4.6%	0.0%
	nncoection	874	51.7%	0.1%
	xonnection	2	0.1%	0.0%
content-length	_ontent-length	3	0.2%	0.0%
	cteonnt-length	39	2.3%	0.0%
	ntcoent-length	40	2.4%	0.0%
	xontent-length	25	1.5%	0.0%
accept-ranges	accept-range	3	0.2%	0.0%
cache-control	cache control	1	0.1%	0.0%
	cache-contol	3	0.2%	0.0%
	cachecontrolheader	1	0.1%	0.0%
content-type	content-typen	0	0.0%	0.0%
	contenttype	1	0.1%	0.0%
last-modified	last modified	4	0.2%	0.0%
	last-modifed	1	0.1%	0.0%
content-location	content location	4	0.2%	0.0%
keep-alive	epke-alive	4	0.2%	0.0%
expires	exires	52	3.1%	0.0%
	expiresdefault	1	0.1%	0.0%
TOTALS		1691	100%	0.0015%

The results show that typos and other alternative spelling mistakes can be largely ignored. The mistake made mostly is misspelling ‘nncoection’ for the HTTP-header ‘connection’. The header was found 874 times, and, although accounting for 51.7% of all misspelled headers, accounts for only 0.0015% of total header properties.

4.3.4.2 File size

A general comparison between actual file size and the file size specified in the Content-Length header can be found in Table 6.

Table 6

General File size Statistics in Bytes (by SPSS)

	Calculated File size	Header Content-Length
N	180640	180640
Mean	228992.46	204758.37
Median	20839.00	8565.00
Sum	41365198871	36987552671
Minimum	0	0
Maximum	716853016	908671000
Std. Deviation	2182386.396	3045190.844

The largest file in the collection is 683.7 Megabytes. The largest content-length as given by the header information is 866.6 Megabytes. The average file size is approximately 223 Kilobytes. The values of the Content-Length headers have an average file size of 200 Kilobytes. The median differs substantially: the header values have a median of 8.4 Kilobytes, while in reality it is 20.4 Kilobytes. A total 226 files have no file size and thus no content, that is only 0.00125%.

4.3.4.3 Encoding

There are three ways to specify the encoding of an XML document[11]. The first option is to use the charset parameter in the Content-Type HTTP-header. An example header looks as follows:

```
Content-Type: text/html; charset=utf-8
```

We have found other headers indicating encoding as well; for example ‘charset’. However, we will not include these headers in this study because they are not part of the HTTP/1.1[12] standard. This means we cannot make sure that those headers had the intention to indicate the encoding of the document, although this is likely. In addition, we also have to take into account the different spelling variations of Content-Type. These alternative spellings can be found in Table 5. Luckily, no contradictory values in alternative spellings of the header have been found on one document. Therefore we can easily include the alternative spellings under this category.

[11] W3C I18n article: Character encodings. http://www.w3.org/International/O-charset.en.php
[12] Hypertext Transfer Protocol -- HTTP/1.1: http://www.w3.org/Protocols/rfc2616/rfc2616.html

A second way to indicate encoding is to use the encoding pseudo-attribute in the XML declaration. An example is:

```
<?xml version="1.0" encoding="utf-8" ?>
```

Also, encoding can be specified in the text declaration at the start of an entity. Currently, we have taken into account the encoding that is specified in the XML declaration only, but in future work we intend to include encoding declarations at the start of an entity as well.

The third is indicates encoding in XHTML using the meta-tag inside the head-element. Identical to the HTTP header, the charset parameter is used again. An example:

```
<meta http-equiv="Content-Type" content="text/html;charset=utf-8" />
```

We extracted all values of the categories above. In total we found 19 different encodings, which are in Appendix 1 – Encodings Found. As is the case with the headers in Table 5, several different spellings of encodings have been found which we have rectified. An overview is available in Table 7.

Table 7

Alternative encoding spellings

Encoding	Alternatives found
utf-8	utf8, uft-8, utf-85
iso-8859-15	iso88859-15
iso-8859-1	8859-1, iso 8859-1, iso_8859-1, iso8859-1, latin1, latin-1
windows-1250	cp1250
windows-1251	cp-1251, cp1251, win-1251
big-5	big5
gb-2312	gb2312
shift_js	shift_js, sjis
tis-620	tis620
ascii	us-ascii
utf-16	utf-15

The header encoding was specified 42,669 times (23.6% of the total collection), the pseudo-attribute encoding 133,983 times (74.2% of the total collection) and the meta tag encoding 16,150 times (8.9% of the total collection).

The visualizations in Figure 15 to Figure 17 show that UTF-8 is most popular, specified respectively 18.8%, 55.3% and 4.6% of XML documents. The pseudo-attribute in the XML doctype

declaration is the only encoding type that was more often specified as utf-8 (55.3%) than not specified (25.8%). The third runner up is ISO-8859-1 in all three encoding types. The figures show only the top 5 of encodings of each type, a complete list can be found in Appendix 1 – Encodings Found.

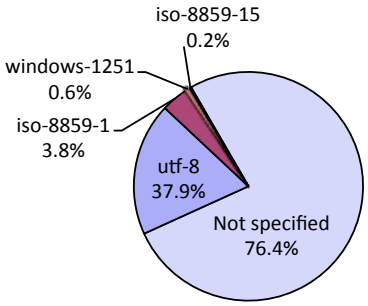

Figure 15. Header Encoding.

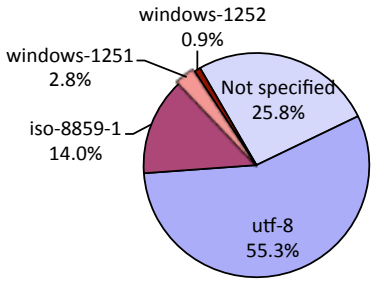

Figure 16. Pseudo Attribute Encoding.

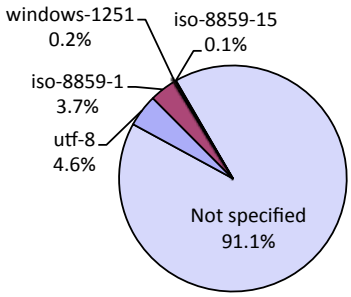

Figure 17. Meta Tag Encoding.

4.3.5 Description of Content Data

4.3.5.1 Schema References

References to schemas is one of the important elements of XML. We focused on the statistics of references to DTDs and XML schemas because they are used most often, and the XML documents include a reference.

4.3.5.1.1 DTDs

DTDs have been downloaded by extracting the system identifier in the XML document header. All external referenced document includes have been downloaded recursively.

Our collection contains 24,426 (13.5%) files with a reference to a DTD. 21,033 (86.1%) of all references use a public identifier, and 24,420 (99.9%) use a system identifier.

We used the system identifier to download the DTDs. A DTD that had already been downloaded, it was not downloaded again. 3059 (12.5%) failed to download via the system identifier. An overview of possible errors in the system identifiers is summarized in Table 8. In the final collection, 21,161 files have a reference to a DTD that could be downloaded.

Table 8

Examples of Errors in References to DTDs

Reference to DTD
c:/users/lakers/styletestout/StyleCheckerReport.dtd
file:/home/paulway/stuff/ISDML/ISDML-1a.dtd
http://95.130.138.2/websidepro/webcatalog/Prodotto.dtd
http://kolloquium.soziologie.ch/Content-Type
http://web.tiscali.it/studioinvernizzi/max.dtd

The DTD schemas contained a total of 5410 includes of other DTDs or entity documents. These have been downloaded recursively, and the original schemas have been modified to include the locally downloaded included schemas. 1786 (33.0%) of them failed to download.

The final collection contains 1375 files downloaded from different system identifier locations. Table 9 shows the ten most referenced, and thus most popular, DTDs. The list is dominated by W3C DTDs, with XHMLT1 DTD on first place. Surprising is that, although only XML was downloaded, HTML DTDs are also referenced. This might be due to the Google and Yahoo query results.

Table 9

Ten Most popular DTDs

Error level	Count	Percentage
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd	8825	41.7%
http://www.w3.org/TR/html4/loose.dtd	3277	15.5%
http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd	2353	11.1%
http://www.w3.org/TR/html4/strict.dtd	1083	5.1%
http://www.w3.org/Math/DTD/mathml2/xhtml-math11-f.dtd	266	1.3%
http://my.netscape.com/publish/formats/rss-0.91.dtd	249	1.2%
http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd	244	1.2%
http://www.w3.org/TR/REC-html40/loose.dtd	192	0.9%
http://www.uni-mannheim.de/mateo/camenatools/DTD/teixlite.dtd	155	0.7%
http://www.wapforum.org/DTD/xhtml-mobile10.dtd	137	0.6%

4.3.5.1.2 XSDs

XSD schemas have been extracted from references in the attribute labels “SchemaLocation” and “noNameSpaceSchemaLocation”. Includes have been downloaded recursively, and the original schemas have been modified to include the locally downloaded versions.

The collection contains 24,087 files with a reference to an XSD (13.3%). There are files that contain multiple references to XSDs. The maximum amount of references in one file is 2399, and 90 documents have more than one reference to an XSD.

Of the unique URLs with XSD schemas, 217 failed to download. The errors in URLs are similar to those of DTDs in Table 8.

A total of 2110 XSD includes were found. Of these, only 23 failed to download. Such a high success rate might be due to the fact that once a DTD is successfully downloaded, the files it includes have a high probability to be available also.

The final collection consists of 437 XSDs from different URLs to XSDs. Table 10 shows the most popular XSDs. This is measured on a different level than most popular DTDs, because files can contain multiple references to separate XSDs. The list in Table 10 is ordered on XSDs which is most referenced in distinct files. The most popular XSD was referred in 19417 different files, which is 82.5% of all files that reference an XSD. In contrast to DTD references, the list is not dominated by W3C schemas, but rather by sitemaps.org, indicating that XSDs are widely used for sitemaps. This can be an interesting fact to services indexing websites or site structures, such as Google.

Table 10

Ten Most popular XSDs

Error level	Count of references in distinct files	%
http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd	19471	82.5%
http://www.google.com/schemas/sitemap/0.84/sitemap.xsd	2456	10.4%
http://www.sitemaps.org/schemas/sitemap/09/sitemap.xsd	236	1.0%
http://www.sitemaps.org/schemas/sitemap/0.9/siteindex.xsd	125	0.5%
http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd	68	0.3%
http://www.xbrl.org/2003/xbrl-linkbase-2003-12-31.xsd	58	0.2%
http://purl.oclc.org/NET/crdo/schemas/archive.xsd	50	0.2%
http://www.loc.gov/standards/mets/mets.xsd	46	0.2%
http://www.google.com/schemas/sitemap/0.84/siteindex.xsd	26	0.1%
http://www.openarchives.org/OAI/2.0/static-repository.xsd	13	0.1%

5. QUALITY ANALYSIS OF DATA

5.1 XML Well-formedness

We created a modified version of the XML parser libxml2. The main changes aim to make the error output in a consistent combination. The original parser would output non-consistent combinations of

error output strings. This made that the translation to our database was difficult. An example of non-consistent error output of xmllint:

```
$ 45.xml:1: parser error : Couldn't find end of Start Tag img line 1
$ 321.xml:633: parser error : Entity 'nbsp' not defined
$ 321.xml:606: parser error : Opening and ending tag mismatch: img line 606 and a
$ 2660.xml:3382: element mod: validity error : ID 2x60 already define
```

Difficulties arise due to the fact that the error-domain (in this case 'parser error' and 'validity error') is not in the same position of every line of output. In addition, specific information like the line number or element name will vary per error, while the type and category of the error will be the same. To categorize the errors we need to separate this specific information. Again, the variations in output formats in combination with a vast amount of different error types make this a difficult task.

After modification of libxml2, the new output always follows the following format:

```
:::FILE:::/scratch/xml/15305.xml
:::LINE:::309
:::ENTITYLINE:::309
:::ELEMENT:::
:::DOMAIN:::validity
:::ERRORLEVEL:::warning
:::FORMATTEDERRORMESSAGECATEGORY:::Attribute already defined
:::FORMATTEDERRORMESSAGESPECIFIC:::Attribute target_version of element CorrespondanceTable
```

This format allows us to parse the output with simple regular expressions and insert the results into our database. Every error encountered by libxml2 consists of:

1) Error level
2) Error domain
3) Error category

Errors in several error categories are then optionally accompanied by:

4) Line number at which the error occurred
5) Element name in which the error occurred
6) Other specific information

The modified version of libxml2 now distinguishes four different error levels; they are summarized in Table 11.

Table 11

Libxml2 error levels

Error Level	Description	Error Level in Original Libxml2
No error	No error.	-- (No output)
Warning	A simple warning.	Warning
Recoverable error	A recoverable error.	Error
Fatal error	A fatal error; the xml can't be parsed.	Error

Before modifications were made, a recoverable error was not distinguishable from a fatal error. When a fatal error is found, the XML cannot be parsed at all. A recoverable error is an error that can be overcome, and were the XML can still be parsed.

The modified version of libxml2 now distinguishes 21 different domains in which errors can occur. These are namespace, XML_FROM_DTD, validity, HTML parser, memory, output, I/O, Xinclude, Xpath, parser, regexp, module, Schemas validity, Schemas parser, Relax-NG parser, Relax-NG validity, Catalog, C14N, XSLT and encoding. New is the XML_FROM_DTD domain (errors in the domain of DTD errors).

Finally, the modified version of libxml2 now categorizes the error that occurred. In our collection a total of 74 categories have been found while the collection was parsed for well-formedness. They can be viewed in Appendix 4 – Overview of libxml2 error categories, accompanied by the error level and error domain. Each error in the parser-domain is always a fatal error. This is as expected: a fatal error occurs when the XML violates well-formedness restrictions, and as a result it cannot be parsed. Similarly, recoverable errors never violate well-formedness restrictions. Recoverable errors occur in the I/O, namespace, parser and validity domain. Lastly, warnings are found in the parser as well as the namespace domain.

The new output format automatically makes a distinction between an error message category and the extra information about the error specific to the document.

The analysis outputs a total of 3,181,824 errors. The distribution across error levels is shown in Table 12.

Table 12

Distribution of errors on Error level during well-formedness parsing

Error Level	Count	Percentage
No error	827	0.03%
Warning	5808	0.18%
Recoverable Error	883231	27.76%
Fatal Error	2291958	72.03%

Fatal errors are the most serious errors, they account for 72.0% of all errors. When fatal errors occur, an XML document is practically unusable. We will therefore provide a detailed description of the fatal errors found in our collection in the next section. In addition, we will give a short analysis of recoverable errors and warnings accordingly.

5.1.1 Fatal errors

A total of 2,291,958 fatal errors (72% of all errors) were found in the collection. They occurred in a total of 26,377 different files (14.6% of total collection). That means that a stunning 154,263 (85.4%) documents in the collection is well-formed, as is shown in Figure 18.

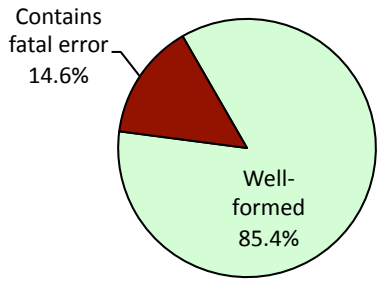

Figure 18. Percentage of Well-formed XML Documents in Collection.

Figure 19 shows that, of all errors fatal errors, 'Opening and ending tag mismatch' is the most encountered; 945,872 times (41.3% of all fatal errors). Second is 'premature end of data' with 553,620 times (24.2%) and "EntityRef: expecting ';' " occurred only 253,911 times (11.1%).

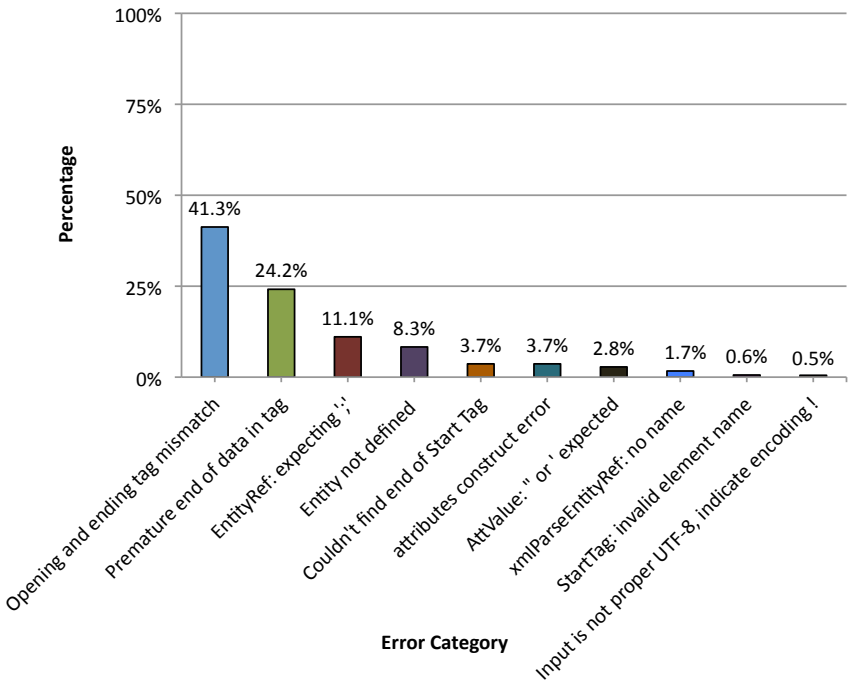

Figure 19. Top Ten Fatal Error Categories in XML document parsing.

More interesting is to check which errors occur in the most separate documents. Obviously, a solution to those errors will have an effect on most of the documents in the collection. Figure 20 shows that this list does not differ a lot from Figure 19. 'Opening and ending tag mismatch' is also encountered most often in different documents (16,996 different documents) and the second runner up is also the same: 'Premature end of data in tag' (14,250 different documents). Third is an unknown encoding (11,615 different documents). This last error does not necessarily have to be a fatal error, as libxml2 allows specifying the encoding of a document manually. We will analyze encoding more thoroughly in section 5.2.

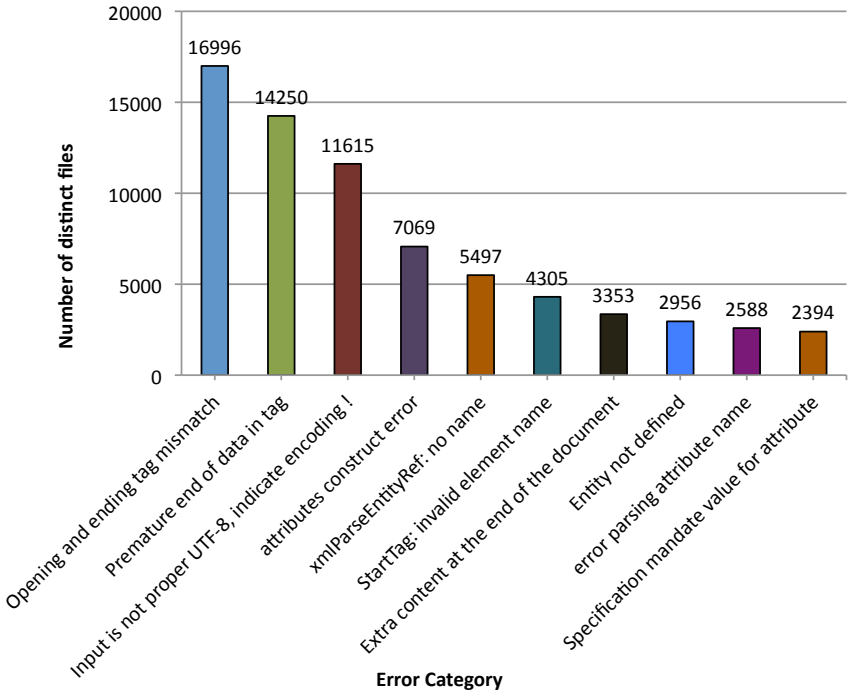

Figure 20. Top Ten Fatal Error Categories in XML document parsing in distinct files.

Furthermore, we can say that the distribution of errors across error categories follows a Pareto's Principle. The Pareto Chart in Figure 21 shows the first nine error categories. Approximately 20% of error categories (a total of 74) account for 99% of the fatal errors.

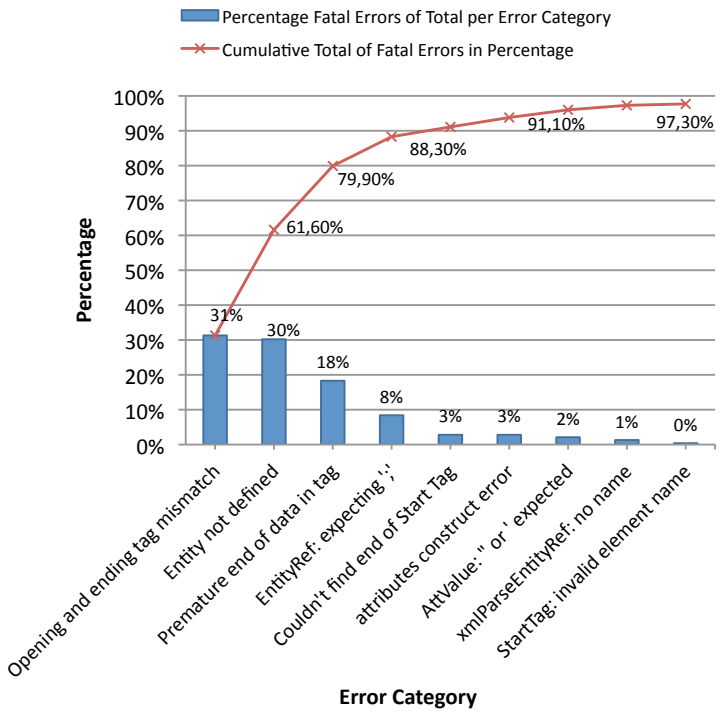

Figure 21. Pareto Chart of Fatal Errors per Error Category.

The amount of fatal errors found per document differs. Generally, a lot of files have a small amount of fatal errors, and only a small amount of files have a large amount of fatal errors. Indeed, the highest percentage of documents contains only one fatal error: 5708 documents (21.6% of all documents with a fatal error), followed by the second highest percentage of documents containing two fatal errors: 1271 documents (4.8% of all documents containing a fatal error). The top 5 is provided in Table 13.

Table 13

Top Five of Amount of Fatal Errors found in Documents

Amount of Fatal Errors	Amount of Documents containing that Amount of Fatal Errors	Percentage as opposed to All Documents containing at least One Fatal Error
1	5708	21.6%
2	1271	4.8%
3	539	2.0%
4	663	2.5%
5	468	1.8%

We are curious which fatal errors occur solely in one document. Figure 22 shows that 32.4% of single fatal errors in a document occur because the document uses an unknown encoding type. As said

above, this might not necessarily be a fatal error in the document, but rather an option of libxml2 that needs specification. This means that the actual amount of well-formed documents might be even higher.

The fatal error solely responsible for failure to adhere to well-formedness constraints is 'Opening and ending tag mismatch', accounting for 25.1%. This might be caused by the fact that ending tags are not properly closed or not properly nested.

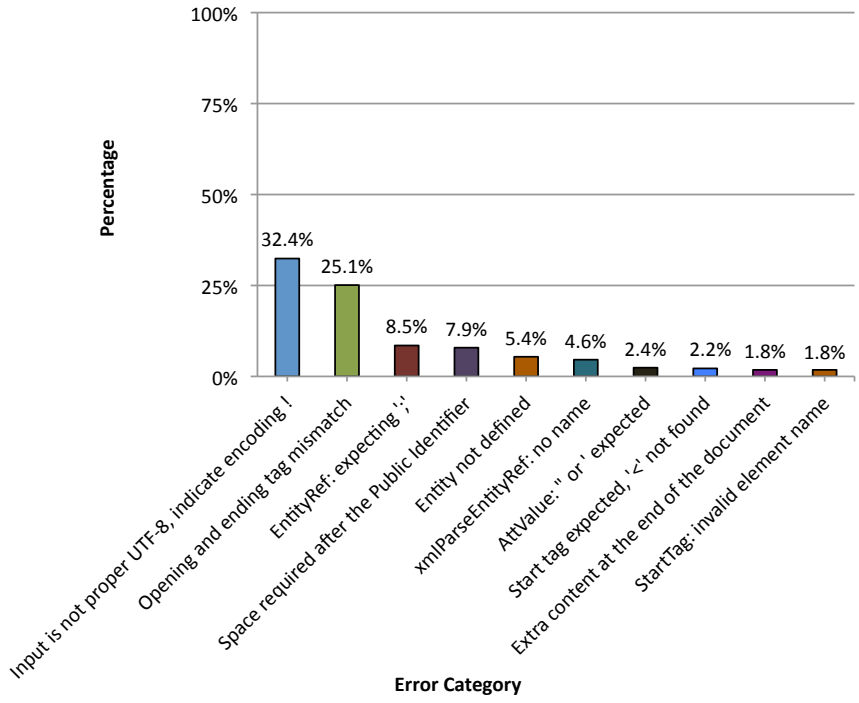

Figure 22. Top Five Errors Solely Responsible for Well-formedness Failure.

Where do the documents with fatal errors come from? Figure 23 shows the distribution of files with fatal errors across the zones defined in Table 4. In comparison with the original distribution of all documents in Figure 13 there is not much difference: the categories '.com', EU and Rest of the World rank highest on the chart. We therefore must conclude that fatal errors in documents occur in every region, and have approximately the same distribution.

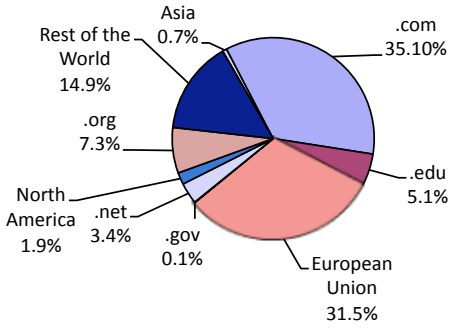

Figure 23. Distribution of Files With Fatal Errors.

In addition, can the file size of a document be used to predict whether a document will contain a fatal error? Very large documents could have more room to contain a fatal error than small documents. We used a linear regression analysis, and for each document in the collection we selected the calculated file size, and whether or not it contained a fatal error. The two are indeed statistically significant, LR $\chi^2(1, N = 180414) = 1280.19$, $p < .01$. Furthermore, the file size does also indeed seem to be a statistically significant predictor of a file to contain a fatal error ($Wald = 576.40$, $df = 1$, $p < 0.001$).

If we take it one step further, is there also a relation between the file size of a document and the amount of fatal errors that occur in it? One could expect that a larger document should contain more fatal errors than a small one. We selected the file size and amount of fatal errors of all files that contain fatal errors. Because both are not normally distributed[13], we use Spearman's rho. Table 14 shows that they are indeed correlated with $r(26154) = .356$, $p < .01$. The coefficient of .356 however, does indicate that there are several other factors that play a role in the amount of fatal errors in a document.

[13] Kolmogorov-Smirnov test. Sig value < 0.05.

Table 14

Correlation of Amount of Fatal Errors to File size

Correlations

			amountoffatalerrors	calculatedfilesize
Spearman's rho	amountoffatalerrors	Correlation Coefficient	1.000	.356**
		Sig. (2-tailed)	.	.000
		N	26156	26156
	calculatedfilesize	Correlation Coefficient	.356**	1.000
		Sig. (2-tailed)	.000	.
		N	26156	26156

**. Correlation is significant at the 0.01 level (2-tailed).

5.1.2 Recoverable Errors

Recoverable errors are those errors that do not violate the well-formedness constraints of XML, but still produce an error. There were a total of 883,231 (27.76% of all errors) recoverable errors found.

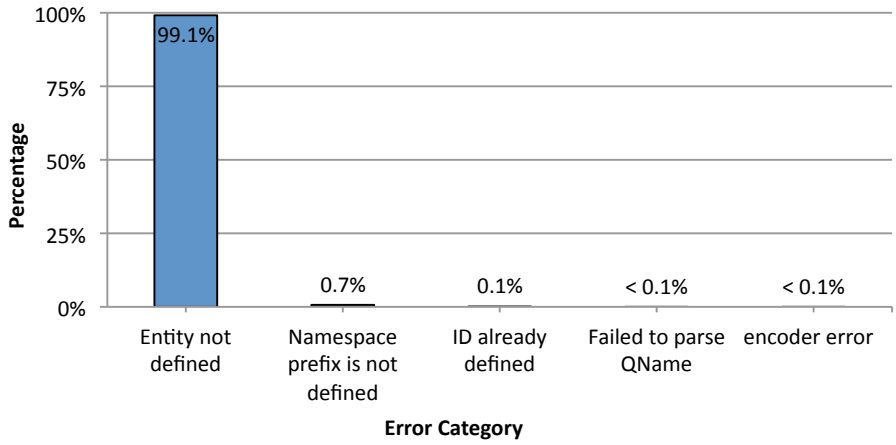

Figure 24. Top Five Recoverable Error Categories in XML well-formedness.

As can be seen in Figure 24, over 99% of recoverable errors are that an entity is not defined. These are   (555,571 times, 63.5% of total), é (65,107 times, 7.4% of total) and ó (24,488 times, 2.8% of total). Indeed, these are not part of XML which only supports 5 entities by default: & (&) < (<), > (>), " (") and ' ('). However, they are part of XHTML specifications. It is possible that the nbsp errors occur in XHTML documents that would normally support them. The analysis of validation in section 5.3.1.1 indeed shows that not defined entities account for only 1.3% of Recoverable Errors during validation with DTDs.

5.1.3 Warnings

A total of 5808 warnings were encountered during XML parsing, accounting for only of total 0.18% warnings and errors output. The error 'xmlns: URI is not absolute' accounts for 96.9% of total warnings.

5.2 Encoding

We made an assessment of the quality of the encodings specified. In other words: we checked whether the indication of the encoding the documents used does in fact correspond with the encoding used by the document.

A document can comply with multiple encodings because certain encoding types overlap. For example, all characters in the ASCII character set can be expressed by the ISO-8859-1 character set, and all characters in the ISO-8859-1 character set can be expressed in the UTF-8 set. This can be written as follows:

$$\text{ASCII} \subset \text{ISO-8859-1} \subset \text{UTF-8}$$

This means that, for certain documents, it can be ASCII compliant as stated in the Content-Type HTTP-header, and also UTF-8 compliant as stated in the document type declaration. But it also allows that certain documents are UTF-8 compliant as stated in the Content-Type HTTP-header, but are *not* ISO-8859-1 compliant as stated in the document type declaration.

Of every document in the collection, we checked whether the three specified encodings as pointed out in paragraph 4.3.4.3 (Encoding) were compliant with the document. Reliable character encoding checking is difficult. To attain reliable results of this analysis, we have chosen to evaluate only those character encoding sets for which a reasonable reliable checking is available. These sets are summarized in Appendix 2 – Encoding Detection methods together with the used checking algorithm.

When we check overall results, without making any distinction where the encoding was specified, 99.47% of all specified encodings is correct. Only 0.53% is incorrect. Figure 25 shows visualizes this is an enormous percentage.

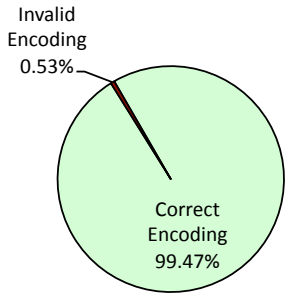

Figure 25: Overall Valid and Invalid Encoding Specification.

Which character sets are always correctly specified and which are always wrong? Figure 26 shows the results per Encoding Type (The data is also available in table format in Appendix 2 – Encoding Detection methods).

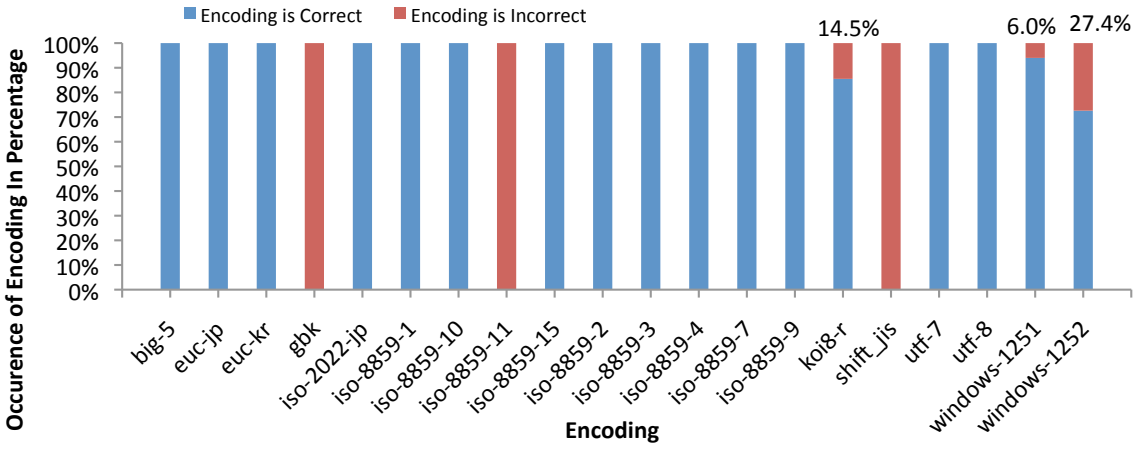

Figure 26. Encoding and Incorrect Encoding per Encoding Type.

We see that UTF-8, UTF-7, EUC-KR, EUC-JP, BIG-5 and ISO-2022-JP are always correctly specified (100%). In addition, all ISO-8859-x standards are always specified correctly, with the only exception being ISO-8859-11, which is always specified incorrectly. GBK and SHIFT_JIS are also specified incorrectly in all cases. A system can rely on the 100% correctly specified encodings. With the 100% negative specified encodings, it always knows the encoding is not correct. It becomes more problematic if certain encodings are sometimes specified correctly and sometimes incorrectly, as there is, in contrast to the 100% specified correctly or incorrectly encodings, no information to gather from it. This is the case for Windows-1251, Windows-1252 and KOI8-R, which are specified incorrectly respectively 6.0%, 27.4% and 14.5%.

If we split up the results per encoding type, results per type are very similar to the overall results, as Table 15 shows. When specified, the XHTML meta tag is most reliable, followed by the

encoding attribute in the doctype declaration, and lastly the HTTP header, but the differences are extremely small. We omit showing the individual counts of specified encodings because they are very similar to the general ones.

Table 15

Encoding quality of XML documents per encoding type

	Content-Type HTTP header	Attribute in Document Declaration	Meta-Tag
Validates	99.43%	99.46%	99.60%
Lies	0.57%	0.54%	0.40%
Total	100.00%	100.00%	100.00%

Furthermore, we can ask how many encodings in the document declaration that is faulty is corrected by an always correct encoding type. Table 16 shows all combinations of contradictory encoding values.

Table 16

Contradictory encoding values

Header Encoding		Pseudo Attribute Encoding		Meta Tag Encoding		Count
Encoding	Correct	Encoding	Correct	Encoding	Correct	
koi8-r	true			windows-1251	false	2
windows-1251	false	utf-8	true			68
koi8-r	false	utf-8	true			6
windows-1251	false	iso-8859-1	true	windows-1251	false	1
iso-8859-1	true	windows-1252	false			1
utf-8	true			windows-1252	false	2
windows-1252	false	iso-8859-1	true			1
utf-8	true	windows-1251	false			5
iso-8859-1	true	windows-1251	false			3
windows-1251	false	utf-8	true	utf-8	true	1
utf-8	true			windows-1251	false	1
koi8-r	false			iso-8859-1	true	3
koi8-r	false	iso-8859-1	true	iso-8859-1	true	10
windows-1251	true	windows-1252	false			1
utf-8	true	windows-1252	false			1
utf-8	true	shift_jis	false			2
windows-1252	false	iso-8859-15	true			1
iso-8859-1	true	windows-1251	false			1
utf-8	true	windows-1252	false			2
iso-8859-1	true			windows-1252	false	1
windows-1251	false			utf-8	true	1
shift_jis	false			utf-8	true	1
koi8-r	false	iso-8859-1	true			2

It shows that it is always the case, except for documents were a combination of koi8-r, windows-1251 and koi8-r is specified, but this occurs at only 3 documents (0.00167% of total collection). If one would use the proposed hierarchy of encodings and all three encoding types, a correct document encoding of almost 100% could be reached.

5.3 XML Validity

XML documents can be compared to match constraints of XML schema languages. If the document matches the constraints, the document is said to be *valid*. If the document does not match the constraints, it is said to be *invalid* (Harold, 2001). However, XML files can only be valid if they are

well-formed. In the previous paragraph we have evaluated which files in the collection are well-formed. In addition, XML files can only be valid if the schema it references complies to standards itself. So the first step in the validation of the XML documents is to find out which schemas in the collection can be compiled.

The second step consists out of validating the well-formed XML files that reference a schema that can be compiled. We will discuss the errors and categories found of the documents that failed to validate. Furthermore, we will try to correlate XML document properties and schema properties with the fact of a document will validate or not.

We will split the analysis between files that reference a DTD and files that reference an XSD schema.

5.3.1 DTDs

Because of its syntax, there is no standard way to validate DTDs (Sahuguet, 2001). To find out if a DTD could be compiled, we tried to validate a simple XML file with the DTD using libxml2:

```
xmllint --noout --dtdvalid dtdtovalidate.dtd simple.xml
```

The content of the simple XML file was:

```
<root />
```

When the DTD could compile, libxml2 would output that the simple XML document is not valid to the schema, or output that it is valid. Otherwise, it could not compile.

Of all DTD schemas that could be retrieved, 909 (66.1%) can be compiled. Table 17 contains an overview of the numbers and percentages of all files with a reference to a DTD. Of all well-formed documents, 6305 contain a reference to a DTD schema that can be compiled (29.8% of all documents with a reference to a DTD, and 88.6% of all well-formed documents with a reference to a DTD).

Table 17

Overview of DTDs compilation numbers

		DTDs		
		Compiles	Fails to Compile	
XML documents	Well-formed	6305 (29.8%)	815 (3.9%)	7120 (33.6%)
	Not Well-formed	8934 (42.2%)	5107 (24.1%)	14041 (66.4%)
		15239 (72.0%)	5922 (28.0%)	21161 (100%)

Is there a relation between the well-formedness of a document, and the fact whether its DTD schema can be compiled? Using binary logistic regression, it can be shown that there is indeed a relationship between the two variables: $\chi^2(1, N = 21161) = 1456.38$, $p < .01$. Well-formed XML documents have a chance of 88.6% of referencing a DTD that can be compiled, and XML documents that are not well-formed only have a chance of 63.6%.

If we look at the geographic distribution of documents that reference DTDs that can be compiled or that fail to compile, as shown in Figure 27 and Figure 28, we see that in both cases the distribution is again in compliance with the overall geographic distribution of documents. The zones '.com', EU and Rest of the World have the most documents that contain a reference to a DTD that both compiles and fails to compile. Worth a note is the fact that the '.edu' domain contains 11.1% of documents that reference a DTD that can be compiled.

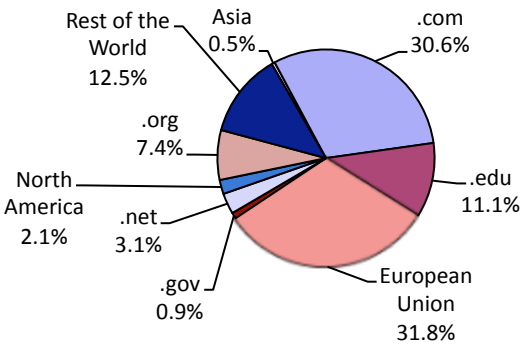

Figure 27. Geographic Distribution of Documents referencing a DTD that can be compiled.

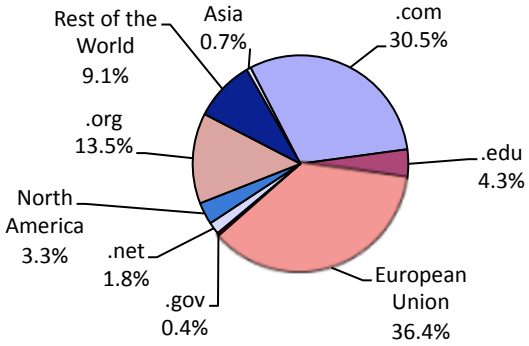

Figure 28. Geographic Distribution of Documents referencing a DTD that fails to compile.

When a schema cannot compile, several errors are outputted. This tells us what is wrong with most DTDs and XSDs. We will not discuss this here, but have a separate section on the validity of DTD and XSD schemas.

5.3.1.1 Validating XML with DTDs

In this step we validate the XML documents from the previous paragraph that are well-formed and reference a DTD schema that can be compiled.

Of the 6305 well-formed XML files with a reference to a DTD that can be compiled, 2046 (32.5%) documents validates, consequently 4259 (67.5%) fail to validate. That means that 8.4% of the documents that specify a DTD, validates with it.

Overall 1,857,093 errors have been found during validation. The distribution of error levels can be seen in Table 18.

Table 18

Distribution of errors on Error level after validating with DTDs

Error Level	Count	Percentage
No error	800	0.0004%
Warning	1507	0.0008%
Recoverable Error	1,821,610	98.1%
Fatal Error	33,176	0.018%

We see that there are 33,176 fatal errors encountered during validation that were previously not found. These include 'Opening and ending tag mismatch' (14,476) and occur in 406 separate documents. For the rest of this paper we will assume that these errors are validation errors, but we will note that in further research a more robust parser should be used.

We are most interested in recoverable errors, because this is the domain where validation errors are encountered. A total of 28 different errors have been found, of which the top ten is shown in Figure 29.

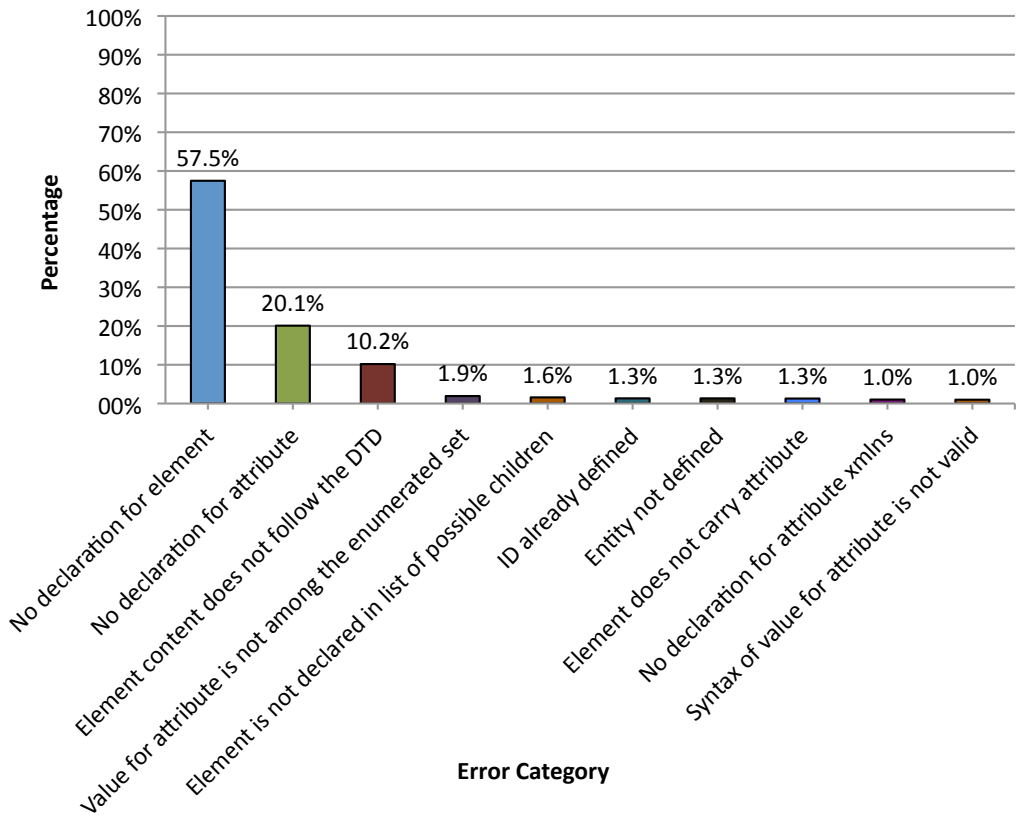

Figure 29. Top Ten Recoverable Error Categories in DTD validation based on count of error category.

Figure 29 also shows that the error that is most encountered, is not necessarily encountered in the most number of files. We see that the error 'No declaration for attribute' is the second most encountered error (20.1%), but is encountered across the largest number of distinct files. However, Figure 30 shows that the errors that occur the most are in general also the ones that occur in the largest amount of distinct documents.

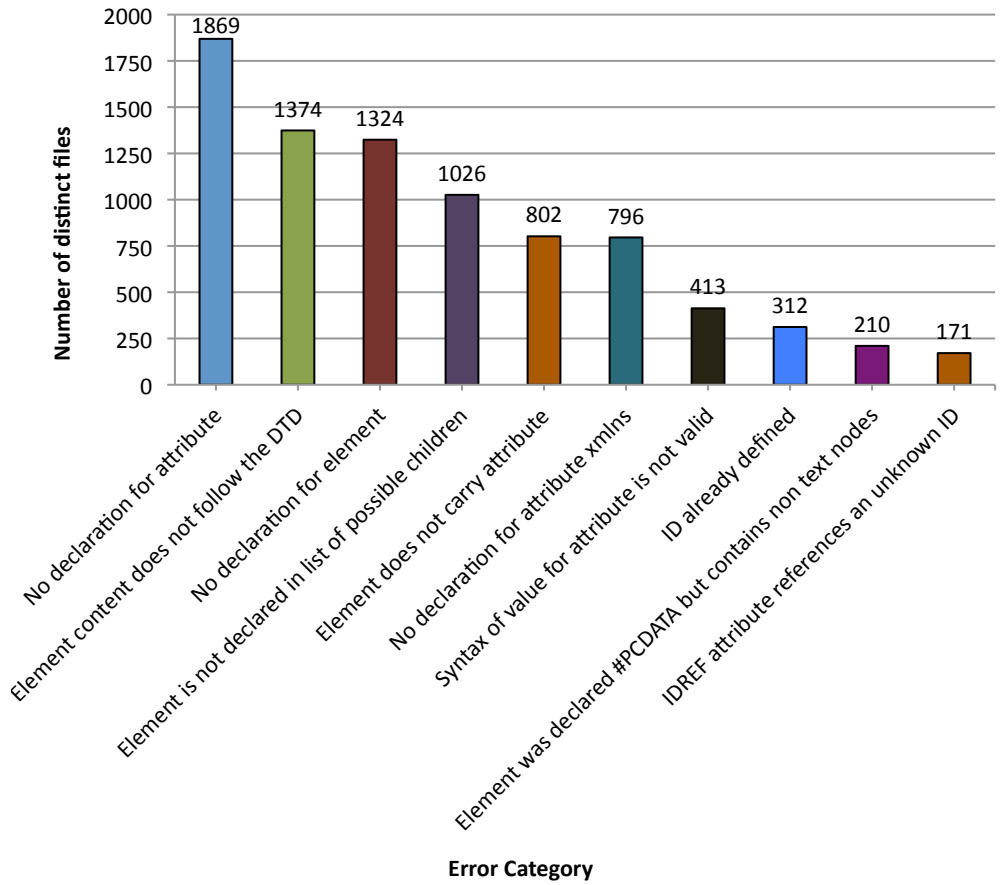

Figure 30. Top Ten Recoverable Error Categories in DTD validation based on occurrence in files.

Let's take a closer look at the top three validation errors. The elements that miss declaration most often are DEFINITION (60,691 times in 26 different files), memorial (60,536 in 31 different files), mrow (57,260 in 152 different files). Analyzing the error 'No declaration for attribute' does not give us much information because the information is extremely specific. For example, the attribute 'available' on element 'offer' is most often not declared (82,158 in 26 different files). The third validation category 'Element content does not follow the DTD' has a lot of errors concerning CDATA. CDATA is encountered while another element is expected. For example:

```
Element units: expecting (unitless | unit+), got (CDATA)
```

The accompanied snippet of XML is:

```
<field>
    <name>---</name>
    <definition>Number 5</definition>
    <units>---</units>
</field>
```

This leads us to ask which DTDs, out of all the files that contain a validation error, are referenced the most. A total of 444 DTDs are referenced in files that have validation errors. Table 19 shows the top five DTDs that are referenced.

Table 19

Top Five DTDs in which most validation errors and warnings occur

DTD (Reconstructed URL)	Count	%
http://www.editeur.org/onix/2.1/reference/onix-international.dtd	35,1601	18.9%
http://www.cs.washington.edu/research/projects/xmltk/xmldata/data/nasa/dataset_053.dtd	21,1074	11.4%
http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd	160,483	8.6%
http://www.mortsdanslescamps.com/general_fichiers/deportation.dtd	101,716	5.5%
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd	95,361	5.1%

Again, it is interesting to see which DTDs have errors and warnings across the largest number of different files. The list shown above can, for example, contain one document with a huge number of errors. The top five DTDs with which most errors occur in different XML documents is shown in Table 20.

Table 20

Top Five DTDs responsible for most errors and warnings in distinct number of documents

DTD (Reconstructed URL)	Is referenced in distinct number of documents containing validation errors	%
http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd	1405	38.3%
http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd	606	16.5%
http://www.w3.org/Math/DTD/mathml2/xhtml-math11-f.dtd	181	4.9%
http://www.w3.org/TR/MathML2/dtd/xhtml-math11-f.dtd	89	2.4%
http://www.wapforum.org/DTD/xhtml-mobile10.dtd	64	1.7%

As can be seen, the list is once again dominated by W3C DTDs. When one compares this list with Table 9 there is a lot of similarity. It could be possible that these DTDs contain the most validation errors simply because of the fact that the largest number of documents references them. However, it could also be possible that these DTDs contain the most validation errors because they are complex to comply with.

5.3.2 XSDs

In contrast with DTDs, XML Schema is in fact XML, so it can be checked using the same tools (Sahuguet, 2001). Again, we used libxml2 to check if an XML Schema is well-formed and can be compiled:

```
xmllint --noout --schema xsdtovalidate.xsd simple.xml
```

The contents of the simple XML file were:

```
<root />
```

When the XSD could compile, libxml2 would output that the simple XML document is not valid to the schema, or output that it is valid. Otherwise, it could not compile.

Of all XSD schemas that could be retrieved, 273 (62.5%) can be compiled. Table 21 shows that 28,338 of all well-formed documents contain a reference to an XSD that can be compiled (86.3% of all well-formed documents with a reference to an XSD). Another interesting fact is that almost all documents that have a reference to an XSD, is well-formed (99.1%).

Table 21

Overview of XSDs compilation numbers

		XSDs		
		Compiles	Fails to Compile	
XML documents	Well-formed	20365 (86.3%)	3010 (12.8%)	23375 (99.1%)
	Not Well-formed	166 (0.7%)	52 (0.2%)	218 (0.9%)
		20531 (87.0%)	3062 (13.0%)	23593 (100%)

The distribution per zone of XSDs that can be compiled and that fail to compile is shown in Figure 31 and Figure 32.

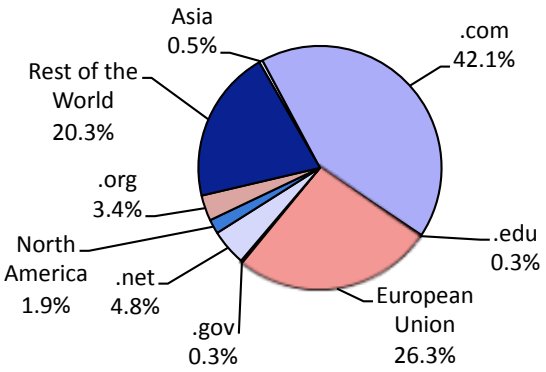

Figure 31. Geographic Distribution of well-formed XML Documents referencing an XSD that can be compiled.

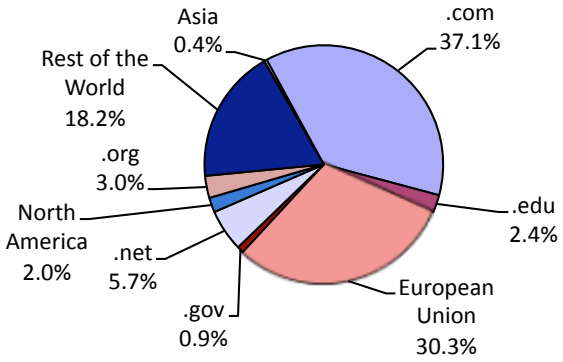

Figure 32. Geographic Distribution of well-formed XML Documents referencing an XSD that fails to compile.

In both cases the documents are again evenly distributed across the geographic regions and top level domains, which indicates that XML documents from all contain references to correct and incorrect XSDs.

5.3.2.1 Validating XML with XSDs

In this step we validate the XML documents that are well-formed and reference an XSD schema that can be compiled.

13,950 (68.5%) documents validated with their XSD, and 31.5% failed to validate.

In all, 14,304,409 errors have been found during validation. The distribution over error level is shown in Table 22.

Table 22

Distribution of errors on Error level after validating with XSDs

Error Level	Count	Percentage
No error	8	0.0001%
Warning	431	0.0030%
Recoverable Error	14,303,940	99.9967%
Fatal Error	30	0.0002%

A total of 26 different error categories were found in the recoverable errors. As with DTDs, we are interested in the Recoverable errors as validation errors occur in this category. Figure 23 shows the top five of the most encountered errors during validation with XSDs. From the 6th onwards, the percentages are 0.1% and less.

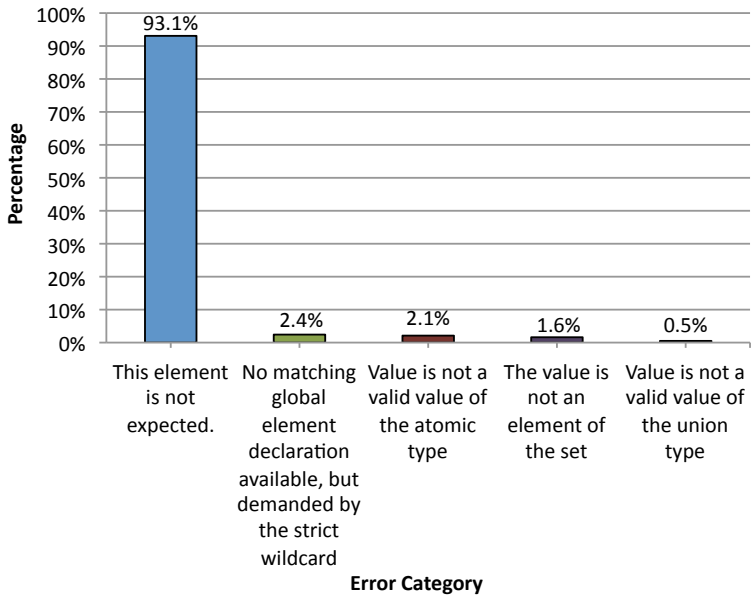

Figure 33. Top Five Recoverable Error Categories in XSD validation based on count of error category.

'This element is not expected' is encountered 13,314,353 times (93.1% of total recoverable errors). The second most encountered recoverable error 'No matching global element declaration available, but demanded by the strict wildcard', is encountered 347,116 times (2.4% of total recoverable errors).

It is again more interesting to see which of the recoverable errors occur in the most distinct files. Obviously, these errors affect the largest number of documents. Figure 34 shows that the list is pulled by the same error: 'This element is not expected', followed by 'Value is not a valid value of the atomic type'.

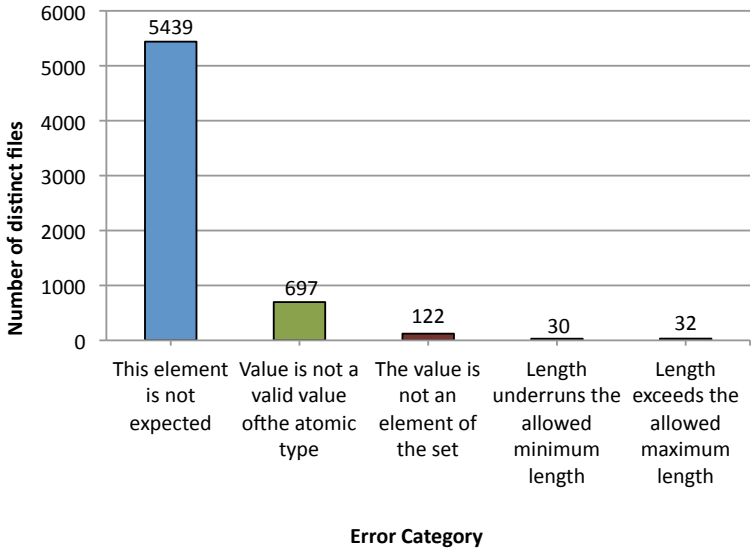

Figure 34. Top Five Recoverable Error Categories in XSD validation based on occurrence in files.

When we look at the number of files that reference an XSD and contain a validation error, sitemaps.org XSDs are referenced by the largest number of documents with validation errors (see Table 23). As is the case with DTDs, this might be due to the fact that these XSDs are simply referenced most often in the collection, as well as due to the fact that the XSDs are complex to work with and are highly susceptible to mistakes.

Table 23

Top Five XSDs responsible for most errors and warnings in distinct number of documents

XSD (Reconstructed URL)	Count	%
http://www.sitemaps.org/schemas/sitemap/0.9/sitemap.xsd	5974	93.1%
http://www.sitemaps.org/schemas/sitemap/09/sitemap.xsd	85	1.3%
http://www.openarchives.org/OAI/2.0/OAI-PMH.xsd	47	0.7%
http://graphml.graphdrawing.org/xmlns/1.0/graphml.xsd	2	0.0%
http://assault.cubers.net/docs/schemas/cuberef.xsd	1	0.0%

5.3.3 Predicting valid XML documents

An XML document is most valuable when it validates to its referenced schema. When it validates it can be processed directly by applications (Sahuguet, 2001). In the previous paragraphs we have seen that the amount of XML documents on the web that validates with a referenced schema is very low. The Venn diagram in Figure 35 shows that this is the case for only 9.2% of documents on the web.

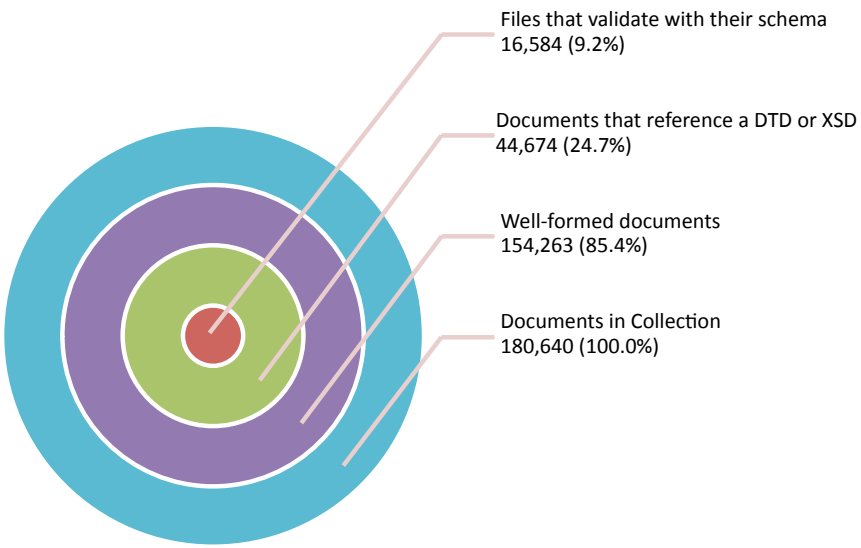

Figure 35. Venn-diagram of All XML documents to those validating with their referenced schema.

If an application could know by forehand whether an XML document on the web would not validate with its referenced schema, one could prevent errors and other unforeseen behavior of applications. In other words, are there properties by which an application can predict whether or not a document will validate? We have made a selection of a document's properties that can be useful in predicting its validity. Several properties that one would expect to be a good predictor are not attractive because they are computationally expensive, or can only be used on well-formed XML data. For example, the amount of recursive elements in a DTD might well be a predictor of complexity of a document, which in turn could affect the probability that a document will validate. However, this can only be used on well-formed XML documents and is therefore of limited use since XML documents from the web are not all well-formed. The selected properties are shown in Table 24.

Table 24

Properties of documents on which to predict validation

Property	Type	Description
Domain extension	Nominal	The domain name extension
Content-Length	Interval	Length of the document as specified in the HTTP 1.1 Header Field
Content-Type charset	Nominal	The encoding as specified in the HTTP 1.1 Header Field
Server	Nominal	IIS or Apache server as specified in the HTTP 1.1 Header Field

The Content-Length HTTP header is an interval variable, but not normally distributed[14]. We use spearman's rho to determine if there is a relationship between the Content-Length HTTP header and the fact if the document validates. There indeed is a relationship, but many other factors play a role. The two are indeed statistically significant, $r(131{,}831) = .214$, $p < .01$.

Regarding base domain, we did a binary logistic regression analysis. The produced model does indicate that domain name extension is statistically significant and explains variations in validity of the documents (χ^2=6087.791, df=334, $p < 0.01$). The model has an overall success rate of 75.8% of validity that was predicted correct; with only the constant this is 75.3%. Which domain name extensions have significant influence? Table 25 provides an overview of domain name extensions that are found to be statistically significant ($p < 0.05$); a total of 33 domain name extensions. It shows that documents from a domain name with extensions '.jp', '.org.au', '.cat', '.gov.uk', '.gov.br' are .306, .196, .276, .281, .106 times likely to be valid with a referenced schema than to be not-valid valid with a referenced schema. In other words: They are less likely to be valid than to be invalid. All other domain name extension in Table 25 are more likely to be valid than invalid, ranging from 2.226 (.gov) to 24.750 more likely to be valid than invalid (.im).

Also, there are seven domain name extensions in the educational and academic domains (containing .edu or .ac) that are all more likely to be valid than invalid. This might indicate that XML from educational domains are in fact of higher quality.

In contrast, documents from governmental domains in the USA are likely to be valid (.gov), while documents from two other governmental domains are less likely to be valid (.gov.br and .gov.uk).

An other interesting fact is that document from the .uk domain are generally almost 2.5 times more likely to be valid than invalid, while documents from the governmental domain in the uk (.gov.uk) are .281 times more likely (in other words: 3.6 times more likely to be invalid than valid). Is might indicate that documents from the British government are of poorer quality than other documents originating in the UK.

[14] Kolmogorov-Smirnov test. Sig value < 0.05. N = 131,831

Table 25

Domain extensions statistically significant in predicting validity

	B	S.E.	Wald	df	Sig.	Exp(B)	95% C.I.for EXP(B)	
							Lower	Upper
de_ac.at(1)	1.504	.409	13.524	1	.000	4.500	2.019	10.031
de_ac.be(1)	2.890	1.184	5.956	1	.015	18.000	1.767	183.411
de_ac.in(1)	2.293	.666	11.846	1	.001	9.900	2.683	36.527
de_be(1)	1.385	.395	12.317	1	.000	3.996	1.843	8.661
de_biz(1)	.818	.404	4.105	1	.043	2.265	1.027	4.996
de_bz(1)	1.879	.553	11.528	1	.001	6.545	2.213	19.362
de_cat(1)	-1.286	.418	9.484	1	.002	.276	.122	.626
de_com.do(1)	2.603	1.219	4.558	1	.033	13.500	1.238	147.232
de_com.pl(1)	1.812	.471	14.770	1	.000	6.120	2.430	15.416
de_com.tr(1)	2.015	.828	5.917	1	.015	7.500	1.479	38.030
de_edu.au(1)	.894	.421	4.505	1	.034	2.444	1.071	5.578
de_edu.ph(1)	2.890	.882	10.741	1	.001	18.000	3.196	101.382
de_edu.pl(1)	2.299	.506	20.609	1	.000	9.964	3.693	26.885
de_edu(1)	1.834	.392	21.851	1	.000	6.259	2.901	13.504
de_fm(1)	1.210	.446	7.366	1	.007	3.353	1.400	8.033
de_gov.br(1)	-2.245	.639	12.338	1	.000	.106	.030	.371
de_gov.uk(1)	-1.269	.604	4.405	1	.036	.281	.086	.919
de_gov(1)	.800	.399	4.024	1	.045	2.226	1.019	4.866
de_gt(1)	2.420	.923	6.870	1	.009	11.250	1.841	68.739
de_ie(1)	1.067	.409	6.799	1	.009	2.906	1.303	6.478
de_im(1)	3.209	.862	13.845	1	.000	24.750	4.566	134.162
de_jp(1)	-1.186	.500	5.626	1	.018	.306	.115	.814
de_kr(1)	2.315	.717	10.429	1	.001	10.125	2.484	41.266
de_mobi(1)	2.828	.451	39.330	1	.000	16.920	6.990	40.955
de_net.ru(1)	1.077	.513	4.409	1	.036	2.935	1.074	8.017
de_nu(1)	.968	.414	5.473	1	.019	2.633	1.170	5.924
de_org.au(1)	-1.631	.601	7.363	1	.007	.196	.060	.636
de_org.in(1)	2.197	.950	5.348	1	.021	9.000	1.398	57.944
de_pl(1)	1.444	.396	13.309	1	.000	4.239	1.951	9.211
de_se(1)	1.104	.392	7.919	1	.005	3.018	1.398	6.513
de_tk(1)	2.803	.760	13.620	1	.000	16.500	3.723	73.126
de_uk(1)	.913	.394	5.386	1	.020	2.493	1.153	5.391
de_us(1)	1.655	.406	16.605	1	.000	5.231	2.360	11.592
Constant	-1.504	.391	14.807	1	.000	.222		

We choose to analyze the server HTTP header to find an answer on the question if documents server by Microsoft's ISS server were more significantly more likely to serve documents that do validate with a schema they may reference than does Apache's HTTP server. We selected all files that were server by an ISS server and that were served by an Apache HTTP server (N=146,952), and conducted a binary logistic regression analysis. The model is shown in Table 26.

Table 26

Domain extensions statistically significant in predicting validity

	B	S.E.	Wald	df	Sig.	Exp(B)	95% C.I.for EXP(B)	
							Lower	Upper
server(1)	.069	.016	19.535	1	.000	1.072	1.039	1.105
Constant	-1.193	.014	7069.096	1	.000	.303		

The coefficient of server is statistically significant ($p < 0.05$) and indicates that documents server by Apache server have 1.072 times as likely to be valid than documents that are server by Microsoft IIS.

We have addressed the Content-Type HTTP header as follows. We selected all files that had a charset encoding extracted as stated in Section 4.3.4.3 (Encoding). With every file we also stated if it validated with a DTD or XSD schema it referenced. Obviously, if it didn't reference a schema or was not well-formed, it is stated as that it failed to validate. We used a Cramer's V analysis to check for correlation between each encoding and the validity of the documents. The following encodings correlate significantly with the validity of documents:

- iso-8859-1: χ^2 (1, N = 42,669) = 1699.50, $p < .01$, $\Phi_{Cramer} = .200$.
- utf-8: χ^2 (1, N = 42,669) = 1193.11, $p < .01$, $\Phi_{Cramer} = .167$.
- windows-1251: χ^2 (1, N = 42,669) = 47.424, $p < .01$, $\Phi_{Cramer} = .033$.
- koi8-r: χ^2 (1, N = 42,669) = 6.376, $p < .05$, $\Phi_{Cramer} = .012$.
- windows-1252: χ^2 (1, N = 42,669) = 11.763, $p < .01$, $\Phi_{Cramer} = .017$.

For these encodings that correlate, we conducted a binary logistic regression analysis. Table 27 shows that all are statistically significant except for UTF-8.

Table 27

Domain extensions statistically significant in predicting validity

	B	S.E.	Wald	df	Sig.	Exp(B)	95% C.I.for EXP(B)	
							Lower	Upper
iso88591(1)	.937	.099	89.929	1	.000	2.551	2.102	3.096
utf8(1)	-.138	.097	2.028	1	.154	.871	.721	1.053
windows1251(1)	-.470	.124	14.369	1	.000	.625	.490	.797
koi8r(1)	.635	.250	6.434	1	.011	1.887	1.155	3.082
windows1252(1)	-.613	.222	7.630	1	.006	.541	.350	.837
Constant	-.976	.096	103.736	1	.000	.377		

6. CONCLUSIONS

The motivation of this research was based on the fact that while XML collections have been studied statistically, no detailed information about its quality is available as of yet. In this study we have tried to address this gap in the research.

We concluded that existing XML collections were not suitable for this research because they were preprocessed to contain only valid XML documents. We built our own XML collection. The final collection consisted of 180,640 XML documents. Our collection contained a mere 0.007% of duplicate documents. The collection contained documents from 96,650 different Web sites; significantly more than other XML collections. The distribution of Web sites across geographic zones and top-level domains is, however, comparable to other XML collections. The distribution of documents, and the distribution of file size are also similar to other collections. We can conclude that our new collection is well balanced.

Our study showed that the well-formedness of XML documents on the web is surprisingly good with 85.4% of documents being well-formed. The fatal error that was most often encountered was ‘Opening and ending tag mismatch’. The runner up was: ‘Premature end of data in tag’. We have seen that the distribution of errors follows a Pareto’s Principle. Applications that process XML data can benefit greatly by focusing on the errors that occur the most number of times.

The next step consisted of identifying and downloading referenced schema documents. 13.5% of XML documents contained a reference to a DTD. This is a lot less than other XML collections (Barbosa, Mignet, & Veltri, 2005). Furthermore, 13.3% of XML documents contained a reference to an XML Schema, significantly more than in Barbosa’s study (0.09%). It could well be that, since

2003, the use of XML Schema has expanded rapidly to the level of DTDs is now. It seems to be the case that XML Schemas are becoming the schema language of first choice in the XML Web. Nevertheless, the low percentages of schemas can still be seen as evidence to work on techniques to discover semantic information from XML documents (Mignet, Barbosa, & Veltri, 2003). The fact that the referenced DTDs are dominated by W3C DTDs, and the references to XML Schemas are dominated by sitemap.org XML Schemas indicate that a large amount of XML documents is of the same kind, this might help in indicating the most useful directions of research.

Regarding the validation of documents, only 29.8% of documents referencing a DTD, referenced one that could be compiled. Of documents referencing XSDs, 86.3% references an XSD that can be compiled. The low number of compiling DTDs and XSDs can be seen as evidence that work on (semi-)automatically learning DTDs or XML Schemas from XML documents can be extremely useful (Bex, Neven, Schwentick, & Tuyls, 2006). We have seen that most of the validation errors occur because there is an element or attribute used that is not defined in the DTD. The list of DTDs with which the largest number of errors occur in different documents is again dominated by W3C DTDs.

We have seen that, of the well-formed documents that reference a DTD that can be compiled, 32.5% validates. In the case of XML Schema documents, 68.5% validates. It seems to indicate that documents that reference an XML Schema are of higher quality, and thus higher usability, than documents that reference a DTD schema.

Regarding encoding of documents in the collection, we have seen that the encoding of documents was surprisingly correct. Depending on the way of specification, between 99.43% and 99.60% was correct. UTF-8 is the most popular of all types of encoding.

Finally, we analyzed if certain properties of documents were indicators for the fact whether an XML document would be referencing a schema and would validate with that schema. If can be decided whether a document is valid or not before it will be processed, can help in preventing unexpected behavior of documents. We concluded that the Content-Length of the HTTP header was statistically significantly correlated with the fact whether a document would validate. Furthermore, we have concluded that 33 domain name extensions explain variations in validity of the documents, of which five indicate a statistically significant likelihood to contain a document that will not validate, and the rest a statistically significant likelihood to contain a document that in fact will validate. In addition, we have seen that documents that are server by Apache web server are more likely to be valid than documents that are served by Microsoft IIS. Lastly, we have found five charset encodings in the

HTTP Content-Type header that are statistically significantly correlated with the fact if a document validates.

While the web moves towards a more and more computer-to-computer communications (Chen, Hong, & Shen, 2005), the quality of XML documents is of high importance. While we think that the quality of the XML web is not as bad as we expected beforehand, we think that improvements can be made by addressing the most common errors as specified in this study.

7. DISCUSSION

It needs to be said that this study has a number of shortcomings. First, limitations of the data collection could influence the results. The most important limitations of the data collection are:

The collection process is not a simple random sample and therefore it cannot be guaranteed that the collection is a representative sample of the XML Web.

During the collection process only the Yahoo and Google search engines were used. Their indexing and pre- and post-processing could have influenced the collection. In addition, based on a HTML study, these indexes represent only approximately 16% of the whole web (Lawrence & Giles, 2000).

There is only queried on the filename extension 'xml', while other file formats are also XML, for example XSL. On the one hand we miss out on this XML in our collection, but on the other hand we could say that the purpose of that XML has an other purpose than data exchange. Future research could look and compare XML from other extensions.

In addition, the study of the DTD schemas includes only those that could be downloaded using their systems identifier. This means that DTDs that solely use a public identifier are not taken into account. However, 99.9% of DTDs were referenced using a system identifier, so the influence of this on our study will likely be negligible.

Availability of data

The data collection, databases and programs used in this research are available for further research at:

http://mashup0.science.uva.nl/sgrijzen/xmlweb/

We kindly ask to acknowledge the source by mentioning this paper.

References

Abcouwer, T., Gels, H., & Truijens, J. (2006). *Informatiemanagement en -beleid.* Schoonhoven: Academic Service.

Abiteboul, S., & Vianu, V. (1997). Queries and Computation on the Web. *ICDT '97: Proceedings of the 6th International Conference on Database Theory* (pp. 262-275). London, UK: Springer-Verlag.

Abrahamson, E. (1991). Managerial Fads and Fashions: The Diffusion and Rejection of Innovations. *The Acadamy of Management Review , 16* (3), 586-612.

Adelaar, T. (2006). Slides college 9 RMCR. Amsterdam, The Netherlands.

AFAC verzekeringsfietsen. (2010). *AFAC verzekeringsfietsen*. Retrieved May 17, 2010, from AFAC verzekeringsfietsen: http://www.afac-verzekeringsfietsen.nl/html/registreren.html

Ahmed, P. K. (1998). Culture and climate for innovation. *European Journal of Innovation Management* , 30-43.

AI Consulting Ltd. (2010). Retrieved 03 07, 2010, from AI Consulting Ltd & Positive Engagement Through Appreciative Inquiry: http://aiconsult.wordpress.com/

Alter, S. (2008). Service system fundamentals: work system, value chain, and life cycle. *IBM Syst. J. , 47* (1), 71-85.

Alvesson, M. (1990). Organization: From Substance to Image? *Organization Studies , 11* (3), 373-394.

Amabile, T., & Khaire, M. (2008). Creativity and the role of the leader. *Harvard Business Review , 86* (10), pp. 100-109.

Anderson, A. (2008). Going Up? Make Sure Your "Elevator Pitch" Doesn't Let You Down. *Smart Business Matters , 4* (1), pp. 3-4.

Armitage, M. (2006, September 21). The Elevator Pitch. USA.

Avital, M., & Te'eni, D. (2009). From Generative Fit to Generative Capacity: Exploring an Emerging Dimension of Information Systems Design and Task Performance. . *Information Systems Journal , 19*, 345-367.

Azze-Eddine, M., Samia, K.-B., & Douniazed, A. H. (2004). XML-DFG : A Dynamic Forms Generator for XML Valid DTD Document. *RIST , 14* (2), 15-26.

Bailey, J., & Fekete, A. *Eds. ACM International Conference Proceeding Series. 242*, pp. 133-139. Darlinghurst, Australia: Australian Computer Society.

Banea, C., Mihalcea, R., & Wiebe, J. A Bootstrapping Method for Building Subjectivity Lexicons for Languages with Scarce Resources. *LREC 2008.*

Barbosa, D., Mignet, L., & Veltri, P. (2005). Studying the XML Web: Gathering Statistics from an XML Sample. *World Wide Web , 8* (4), pp. 413-438.

Basadur, M., Pringle, P., Speranzini, G., & Bacot, M. (2000). Collaborative problem solving through creativity in problem definition: Expanding the pie. *Creativity and Innovation Management , 9* (1), 54-76.

Bawden, D. (2001, March). Information and Digital Literacies: A Review of Concepts. *Journal of Documentation , 57* (2), pp. 218-259.

Beatty, P., Dick, S., & Miller, J. (2008, Mar/Apr). Is HTML in a Race to the Bottom? A Large-Scale Survey and Analysis of Conformance to W3C Standards. *IEEE Internet Computing , 12* (2), pp. 76-80.

Beckett, D. (1997). 30% accessible - a survey of the UK Wide Web. *Computer Networks and ISDN Systems , 29* (Nos 8-13), pp. 1367-75.

Beerkens, E. (2006). *UvA in top 50, maar...* Retrieved 02 01, 2010, from http://www.scienceguide.nl/print.asp?articleid=104349

Bex, G. J., Martens, W., Neven, F., & Schwentick, T. (2005). Expressiveness of XSDs: from practice to theory, there and back again. *WWW '05: Proceedings of the 14th international conference on World Wide Web* (pp. 712-721). New York, NY, USA: ACM.

Bex, G. J., Neven, F., & Bussche, J. V. (2004). DTDs versus XML Schema: A Practical Study. *WebDB '04, Proceedings of the 7th International Workshop on the Web and Databases* (pp. 79-84). New York, NY, USA: ACM Press.

Bex, G. J., Neven, F., Schwentick, T., & Tuyls, K. (2006). Inference of concise DTDs from XML data. *VLDB '06: Proceedings of the 32nd international conference on Very large data bases* (pp. 115-126). Seoul, Korea: VLDB Endowment.

Boiko, B. (2001). Understanding content management. *American Society for Information Science , 28* (October - November), 8-13.

Boiy, E., Hens, P., Deschacht, K., & Moens, M.-F. (2007). Automatic sentiment analysis of on-line text. *Proceedings of the 11th International Conference on Electronic Publishing*, (pp. 349-360). Vienna, Austria.

Bonaccorsi, & Rossi. (2003). Why Open Source software can succeed. *Research Policy , 32* (7), 1243-1258.

Borden, N. (1964). The Concept of the Marketing Mix. *Journal of Advertising Research* (4), 2-7.

Braman, S. (1989, 09). Defining Information: An approach for policymakers. *Telecommunications policy* , 233-242.

Brooks, R. (2005). Measuring university quality. *The Review of Higher Education , 29* (1), 1-21.

Bulger, M. (2006). *Beyond search: A preliminary skill set for online literacy.* Santa Barbara: Transliteracies Project, University of California.

Catmull, E. (2008, September). How Pixar Fosters Collective Creativity. *Harvard Business Review* .

Chen, B., & Shen, V. Y. (2006). Transforming Web Pages to Become Standard-Compliant through Reverse Engineering. *W4A '06: Proceedings of the 2006 international cross-disciplinary workshop on Web accessibility (W4A): Building the mobile web: rediscovering accessibility?* , pp. 14-22.

Chen, S., Hong, D., & Shen, V. (2005). An experimental study on validation problems with existing HTML web pages. *ICOMP'05: Proceedings of the International Conference on Internet Computing*, (pp. 373-379).

Chesbrough, H., & Spohrer, J. (2006). A research manifesto for services science. *Communications of the ACM , 49* (7), 35-40.

Chesbrough, H., & Spohrer, J. (2006). A research manifesto for services science. *Commun. ACM , 49*, 35-40.

Chesley, P., Vincent, B., Xu, L., & Srihari, R. (2006). *Using Verbs and Adjectives to Automatically Classify Blog Sentiment.* AAAI Spring Symposium Technical Report SS-06-03.

Choi, B. (2002). What are real DTDs like? *WebDB '02, Proceedings of the 5th International Workshop on the Web and Databases* (pp. 43-48). Madison, Wisconsin, USA: ACM Press.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement , 20* (1), 37-46.

Compact Oxford English Dictionary: opinion. (n.d.). (O. U. Press, Producer) Retrieved 06 05, 2009, from Compact Oxford English Dictionary: http://www.askoxford.com:80/concise_oed/opinion

Computable. (2008, 10 03). *Plezier op de werkvloer zorgt voor creativiteit*. Retrieved 03 15, 2010, from Computable.nl: http://www.computable.nl/artikel/ict_topics/loopbaan/2724424/1458016/plezier-op-de-werkvloer-zorgt-voor-creativiteit.html

Cooperrider, D. L., & Whitney, D. (2005). *Appreciative Inquiry: A Positive Revolution in Change* (1st Edition ed.). San Francisco: Berrett-Koehler Publishers.

Corley, K. (2000). The rankings game: managing business school reputation. *Corporate Reputation Review , 3* (4), 319-33.

Cowell, D. W. (1988). New service development. *Journal of Marketing Management , 3* (3), 313-327.

Cremonini, L., Westerheijden, D., & Enders, J. (2008). Disseminating the right information to the right audience: cultural determinants in the use (and misuse) of rankings. *Higher Education* (55), 373–385.

Cullen, R. (2001). Addressing the Digital Divide. *Libraries and Librarians: Making a Difference in the Knowledge Age. Council and General Conference: Conference Programme and Proceedings*, (p. 14). Boston.

Damanpour, F. (1991). Organizational innovation: a meta-analysis of effects of determinants and moderators. *Academy of Management Journal , 31* (3), 555-590.

De Standaard. (2008, 03 21). Toppers trekken andere toppers aan. *De Standaard* .
De Vries, E. (2006). Innovation in services in networks of organizations and in the distribution of services. *Research Policy , 35*, 1037-1051.
De Vries, E. J., & Huizing, A. (2007). Chapter 1 - Information Management: Setting the Scene. In A. Huizing, & E. J. De Vries (Eds.), *Information Management: Setting the Scene.* Amsterdam: Universiteit van Amsterdam.
Denning, P. J. (2004). The social life of innovation. *Communications of the ACM , 47* (4), 15-19.
Devinney, T., Dowling, G., & Perm-Ajchariyawong, N. (2006). *The Business School Rankings Game.* Australian Graduate School of Management Working Paper.
Dienst Infrastructuur Verkeer en Vervoer. (2007). *Het Meerjarenbeleidsplan Fiets 2007-2010.* Amsterdam: DIVV.
Dijk, J. A. (2005). *The Network Society: Social Aspects of New Media.* Sage Publications Ltd.
Dill, D., & Soo, M. (2005). Academic quality, league tables, and public policy: A cross-national analysis of university ranking systems. *Higher Education , 49*, 495-533.
Ding, X., & Liu, B. (2007). The utility of linguistic rules in opinion mining. *Proceedings of the 30th Annual international ACM SIGIR Conference on Research and Development in information Retrieval* (pp. 811-812). Amsterdam, The Netherlands: ACM, New York, NY.
Djellal, F., & Gallouj, F. (2001). Innovation in services: Patterns of innovation organisation in service firms: postal survey results and theoretical models. *Science and Public Policy , 28* (1), 57-67.
Edens, J., Liem, M., Mensink, T., Weve, R., & Zande, L. v. (2006). *Measuring Politics.* University of Amsterdam, Amsterdam.
Edmunds, & Morris. (2000). The problem of information overload in business organizations: a review of the literature. *International Journal of Information Management , 20*, 17-28.
Eindhoven Corpus. (n.d.). Retrieved from Instituut voor Nederlandse Lexicologie - Eindhoven Corpus: http://www.inl.nl/index.php?option=com_content&task=view&id=350&Itemid=579
Eshet-Alkali, Y., & Amichai-Hamburger, Y. (2003). Experiments in Digital Literacy. *Cyberpsychology & Behavior , 7* (4), 421-429.
Esuli, A., & Sebastiani, F. (2006). Determining term subjectivity and term orientation for opinion mining. *Proceedings of the Eleventh Conference on European Chapter of the Association for Computational Linguistics* (pp. 193-200). Trento, Italy: European Chapter Meeting of the ACL. Association for Computational Linguistics.
Fangzhong, S., & Markert, K. (2008). From Words to Senses: a Case Study in Subjectivity Recognition. *Proc. of Coling 2008.* Manchester, UK.
Farrell, E. F., & Werf, M. v. (2007, 05 27). Playing the Rankings Game. *The Chronicle of Higher Education* .
FEM Business. (2008, 10 15). UvA beste Nederlandse universiteit. *FEM Business* .
Field, A. (2009). *Discovering Statistics Using SPSS* (3rd Edition ed.). London: SAGE Publications.
Frei, F. X. (2006). *Commerce Bank.* Harvard Business School.
Furuse, O., Hiroshima, N., Yamada, S., & Kataoka, R. (2007). Opinion sentence search engine on open-domain blog. *Proc. of 20th Int. Joint Conf. of Artificial Intelligence (IJCAI2007).*
Gadrey, J., Gallouj, F., & Weinstein, O. (1995). New modes of innovation: How services benefit industry. *nternational Journal of Service Industry Management , 6* (3), 4-16.
Gallouj, F., & Weinstein, O. (1997). Innovation in services. *Research Policy , 26*, 537–556.
Gels, H., & Abcouwer, A. (1996). *Informatiebeleid in beleidsontwikkelingsprojecten.* Schoonhoven: Academic Service.
Gerritsen, S. (2001). *Schrijfgids voor economen.* Bussum: Coutinho.
Gielissen, T. (2008). *Het ontsluiten van Nederlandse parlementaire publicaties naar Brits voorbeeld.* Bachelor Thesis, University of Amsterdam, FNWI, Amsterdam.
Gilster, P. (1997). *Digital Literacy.* New York: Wiley.
Gioia, D. A., & Corley, K. G. (2002). Being Good Versus Looking Good: Business School Rankings and the Circean Transformation From Substance to Image. *Academy of Management Learning and Education , 1* (1), 107-120.

Google. (2010). *Google WebSearch Protocol Reference for Google Site Search*. Retrieved June 13, 2010, from http://www.google.nl/cse/docs/resultsxml.html#countryCodes

Grewal, R., Dearden, James, A., Lilien, & Gary, L. (2006). Grewal, Rajdeep, Dearden, James, A., Lilien, and Gary, L. (2008). The university rankings game: Modeling the competition among universities for ranking. The American Statistician, 62(3):232-237. *The American Statistician , 62* (3), 232-237.

Grijzenhout, S. (2010). *Exploring a specific technical innovation "the console".* Assignment 1A Information and Innovation, University of Amsterdam, Faculty of Science, Amsterdam Business School, Amsterdam.

Grijzenhout, S., Rheenen, E. v., & Marx, M. (2009). *Er wordt wat afgepraat op het Binnenhof, Proefproject Bachelor Thesis Informatiekunde*. Retrieved 06 14, 2009, from http://student.science.uva.nl/~sgrijzen/verslag_warmup/

Guerrini, G., Mesiti, M., & Rossi, D. (2005). Impact of XML schema evolution on valid documents. *WIDM '05: Proceedings of the 7th annual ACM international workshop on Web information and data management* (pp. 39-44). New York, NY, USA: ACM.

Haan, J. d. (2004). A Multifacated Dyamic Model of the Digital Divide. *IT & Society , 1* (7), 66-88.

Hackett, S., Parmanto, B., & Zeng, X. (2004). Accessibility of Internet websites through time. *SIGACCESS Access. Comput.* , 32-39.

Hargittai, E. (2002). Second Level Digital Divide: differences in people's online skills. *First Monday , 7* (4).

Harold, E. R. (2001). *XML Bible.* New York , NY, USA: John Wiley & Sons, Inc.

Hastbacka, M. A. (2004, December). Open Innovation: What's Mine is Mine... What if Yours Could Be Mine Too? *Technology Management Journal* .

Hecker, F. (1999). Setting Up Shop: The Business of Open-Source Software. *IEEE Software* , 45-51.

Heskett, J. L. (2003). *Southwest Airlines 2002: An Industry Under Siege.* Harvard Business School.

Hippel, E. v. (2001). Perspective: User toolkits for innovation. *Journal of Product Innovation Management , 18*, 247-257.

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, (pp. 22-25). Seattle, WA, USA.

Huizing, A. (2007). Objectivist by default: why information management needs a new foundation. In A. Huizing, & E. d. Vries, *Information Management: Setting the Scene* (pp. 91-110). Oxford: Elsevier Science.

Huizing, A. (2007). The Value of a Rose: Rising above Objectivism and Subjectivism. *Sprouts: Working Papers on Information Systems , 7* (11).

IDEO. (2010, Feb). *Human Centered Design Toolkit.* Retrieved from IDEO: http://www.ideo.com/work/item/human-centered-design-toolkit/

ILPS. (n.d.). *Homepage*. Retrieved 06 03, 2009, from ILPS information and language processing systems: http://ilps.science.uva.nl/

Iyer, B., & Davenport, T. H. (2008, April). Reverse Engineering Google's Innovation Machine. *Harvard Business Review* , pp. 58-68.

Japan Probe. (2007, 5 2). *Disneyland in China?* Retrieved 12 15, 2009, from Japan Probe: http://www.japanprobe.com/2007/05/02/disneyland-in-china/

Jeppesen, L. B., & Frederiksen, L. (2006). Why Do Users Contribute to Firm-Hosted User Communities? The Case of Computer-Controlled Music Instruments. *Organization Science , 17* (1), 45-63.

Jijkoun, V., & Hofmann, K. (2009). Generating a Non-English Subjectivity Lexicon: Relations That Matter. *2th Conference of the European Chapter of the Association for Computational Linguistics (EACL-09).*

Kamps, J., & Marx, M. (2001). Words with Attitude. *1st International WordNet Conference*, (pp. 332-341).

Kamps, J., Marx, M., Mokken, R., & Rijke, M. d. (2004). Using WordNet to measure semantic orientations of adjectives. *Proceedings LREC.*
Kelley, T. (2001). Prototyping is the Shorthand of Design. *Design Management Journal , 12* (3), pp. 35-42.
Kelley, T. (2005). *The Ten Faces of Innovation.* Currency.
Kim, S.-M., & Hovy, E. (2004). Determining the Sentiment of Opinions. *Proceedings of COLING-04*, (pp. 1367-1373). Geneva, Switzerland.
Kim, S.-M., & Hovy, E. H. (2005). Automatic Detection of Opinion Bearing Words and Sentences. *Second International Joint Conference on Natural Language Processing.*
Kirk, J. (1999). Information in organisations: directions for information management. *Information research , 4* (3).
Klettke, M., Schneider, L., & Heuer, A. (2002). Metrics for XML Document Collections. *XMLDM Workshop*, (pp. 162-176). Prague, Czech Republic.
Koehn, N. F., Besharov, M., & Miller, K. (2008). *Starbucks Coffee Company in the 21st Century.* Harvard Business School.
Kolb, D. A. (1984). *Experiential learning: experience as the source of learning and development.* Englewood Cliffs, N.J., USA: Prentice-Hall.
Kosek, J., Kratky, M., & Snasel, V. (2003). Struktura realnych XML dokumentu a metody indexovani. *ITAT 2003: Workshop on Information Technologies Applications and Theory.* High Tatras, Slovakia.
Ku, L., Liang, Y., & Chen, H. Tagging heterogeneous evaluation corpora for opinionated tasks. *LREC 2006.*
Kushal, D., Lawrence, S., & Pennock, D. M. (2003). Mining the peanut gallery: opinion extraction and semantic classification of product reviews. *Proceedings of the 12th international conference on World Wide Web*, (pp. 20-24). Budapest.
Lakhani, K. R., & Hippel, E. v. (2003). How open source software works: "free" user-to-user assistance. *Research Policy , 31*, 923–943.
Landow, G. (1992). Hypertext, metatext and the English canon. In M. Tuman, *Literacy online* (pp. 67-94). Pittsburgh, Pennsylvania: University of Pittsburgh Press.
Language Technologies Institute, Carnegie Mellon University. (2009). *The ClueWeb09 Dataset*. Retrieved 05 06, 2010, from http://boston.lti.cs.cmu.edu/Data/clueweb09/
Lauer, T. (1999). Questions and Information: Contrasting Metaphors. *Association for Information Systems - Americas Conference on Information Systems (AMCIS)* , 640-643.
Lawrence, S., & Giles, C. L. (2000, Spring). Accessibility of information on the Web. *intelligence , 11* (1), pp. 32-39.
Lee, D., & Chu, W. W. (2000). Comparative analysis of six XML schema languages. *SIGMOD Rec. , 29* (3), 76-87.
Leiden University. (2008). *The Leiden Ranking 2008*. Retrieved 02 01, 2010, from http://www.cwts.nl/ranking/LeidenRankingWebSite.html
Lerner, J., & Tirole, J. (2002, Jun). Some Simple Economics of Open Source. *The journal of Industrial Economics* , 197-234.
Levitt, T. (1981). Marketing Intangible Products and Product Intangibles. *Cornell Hotel and Restaurant Administration Quarterly , 22* (2), 37-44.
Liu, B. (2007). *Web Data Mining: Exploring hyperlinks, contents and usage data.* Springer.
Machung, A. (1998, 07-08). Playing the rankings game. *Change , 30* (4), pp. 1-12.
Maes, R. (1999, 04). A Generic Framework for Information Management. *PrimaVera Working Paper 99-03* .
Maes, R. (2003, 06). Informatiemanagement in kaart gebracht. *PrimaVera Working Paper 2003-02* .
Maes, R. (2008, 05). Informatiemanagement of de kunde van het balanceren . *M&O , 3/4*, pp. 332-343.
Maes, R. (2004). Information Management Reconstructed. *PrimaVera Working Paper 2004-19* .

Manning, C., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval.* Cambridge University Press.
Martens, W., Neven, F., & Schwentick, T. (2005). Which XML schemas admit 1-pass preorder typing? *Proceedings of the 10th International Conference on Database Theory* (pp. 68-82). Berlin, Germany: Springer.
Martens, W., Neven, F., Schwentick, T., & Bex, G. J. (2006). Expressiveness and complexity of XML Schema. *ACM Trans. Database Syst. , 31* (3), 770-813.
Marx, M. (2009). Long, often quite boring, notes of meetings. *In Proceedings ESAIR 2009 : Exploiting Semantic Annotations in Information Retrieval.*
McDonough, P. M., Antonio, A. L., Walpole, M., & Pérez, L. X. (1998). COLLEGE RANKINGS: Democratized College Knowledge for Whom? *Research in Higher Education , 39* (5), 513-537.
McDowell, A., Schmidt, C., & Yue, K.-B. (2004). Analysis and Metrics of XML Schema. *SERP '04, Proceedings of the International Conference on Software Engineering Research and Practice* (pp. 538-544). CSREA Press.
McKeever, S. (2003). Understanding Web content management systems: evolution, lifecycle and market. *Industrial Management & Data Systems , 103* (9), 686-692.
McKeown, K., & Hatzivassiloglou, V. (1997). Predicting the semantic orientation of adjectives. *Proceedings of the 35th annual meeting of ACL.*
Meredith, M. (2004). WHY DO UNIVERSITIES COMPETE IN THE RATINGS GAME? AN EMPIRICAL ANALYSIS OF THE EFFECTS OF THE U.S. NEWS AND WORLD REPORT COLLEGE RANKINGS. *Research in Higher Education , 45* (5), 443-461.
Merton, R. (1968). The Matthew effect in science. *Science , 159*, 56-63.
Mignet, L., Barbosa, D., & Veltri, P. (2003). The XML Web: a First Study. *WWW '03, Proceedings of the 12th international conference on World Wide Web. 2*, pp. 500-510. New York, NY, USA: ACM Press.
Miles, I. (2008). Patterns of innovation in service industries. *IBM Syst. J. , 47*, 115-128.
Mishne, G. (2005). Experiments with mood classification in blog posts. *st Workshop on Stylistic Analysis Of Text For Information Access.*
Mlynkova, I., Toman, K., & Pokorny, J. (2006). *Statistical Analysis of Real XML Data Collections (Technical Report).* Charles University, Faculty of Mathematics and Physics, Department of Software Engineering, Prague, Czech Republic.
Morgeson, F. P., & Nahrgang, J. D. (2008). Same as It Ever Was: Recognizing Stability in the BusinessWeek Rankings. *Academy of Management Learning & Education , 7* (1), 26-41.
Mullen, T., & Malouf, R. (2006). A preliminary investigation into sentiment analysis of informal political discourse. *AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*, (pp. 159–162).
Nagelhout, H. (2008). *Rankings en Informatiemanagement: Omgaan met rankings vanuit een informatiemanagement perspectief.* Amsterdam: Universiteit van Amsterdam.
Nicholas, D., & Williams, P. (1998). Review of: P. Gilster. Digital Literacy. *Journal of Documentation , 54* (3), 360-362.
NRC Handelsblad. (2010, 02 01). Toptalent komt niet meer hierheen. *NRC Handelsblad* .
Ofuonye, E., Beatty, P., Dick, S., & Miller, J. (2010). Prevalence and classification of web page defects. *Online Information Review , 34* (1), 160-174.
Osgood, C., Suci, G., & Tannenbaum, P. (1957). *The measurement of meaning.* Urbana, IL: University of Illinois Press.
Osman, D., & Yearwood, J. (2007). Opinion search in web logs. *Proceedings of the Eighteenth Conference on Australasian Database*, *63.* Ballarat, Victoria, Australia.
Osman, D., & Yearwood, J. (2007). Opinion Search in Web Logs. *In Proceedings of ADC.* Ballarat, VIC, AUS.
Osorio, C. A. (2010, April 14). Critical Decisions in Innovation Projects. Amsterdam, The Netherlands.

Pang, B., & Lee, L. (2008). Opinion Mining and Sentiment Analysis. *Foundations and Trends® in Information Retrieval , 2* (1-2), 1-135.
Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing* (pp. 79-86). Association for Computational Linguistics.
Philip, M. (2003). *Noties van Informatiemanagement.* Universiteit van Amsterdam, Faculteit der Natuurwetenschappen en Informatica (FNWI); Faculteit der Economische Wetenschappen en Econometrie (FEE), Amsterdam.
Pietrass, M. (2007). Digital Literacy Research from an International and Comparative Point of View. *Research in Comparative and International Education , 2* (1), 1-12.
Pike, G. (2003). A Comparison of U.S. News Rankings and NSSE Benchmarks. *Association for Institutional Research* , 1-26.
Pine, B. J., & Gilmore, J. H. (1998, July). Welcome to the experience economy. *Harvard Business Review* , pp. 97-105.
Pollach, I., Pinterits, A., & Treiblmaier, H. (2006). Environmental Web Sites: An Empirical Investigation of Functionality and Accessibility. *Proceedings of the 39th Hawaii International Conference on System Sciences.* IEEE.
Provenzo, E. (1992). The electronic Panopticon. In M. Tuman, *Literacy online* (pp. 167-188). Pitssburgh, Pennsylvania: University of Pitssburgh Press.
Raggett, D. (2003). *Clean up your Web pages with HTML TIDY*. Retrieved 04 05, 2010, from http://www.w3.org/People/Raggett/tidy/
Raghavan, S., & Garcia-Molina, H. (2001). Crawling the Hidden Web. *VLDB '01: Proceedings of the 27th International Conference on Very Large Data Bases* (pp. 129-138). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.
Rahm, E., & Do, H. (2000). Data Cleansing: Problems and Current Approaches. *IEEE Data Engineering Bulletin* .
Raymond, E. (1999). *The Cathedral and the Bazaar: Musings on Linux and Open Source from an Accidental Revolutionary.* Sebastapol, CA, USA: O'Reilly and Associates.
Riloff, E., & Wiebe, J. (2003). Learning extraction patterns for subjective expressions. *Conference on Empirical Methods in Natural Language Processing (EMNLP-03)*, (pp. 105–112).
Rogers, E. (1995). *Diffusion of innovations* (4th Edition ed.). New York, NY, USA: Free Press.
Rowley, J. (1998). Towards a Framework for Information Management. *International Journal of Information Management , 18* (5), 359-369.
Sahuguet, A. (2001). Everything You Ever Wanted to Know About DTDs, But Were Afraid to Ask (Extended Abstract). *Selected papers from the 3rd International Workshop WebDB 2000 on The World Wide Web and Databases* (pp. 171-183). London, U.K.: Springer-Verlag.
Saunders, M., Lewis, P., & Thornhill, A. (2004). *Methoden en technieken van onderzoek* (3rd ed.). (P. Smitt, & K. Ulsda, Trans.) Amsterdam: Pearson Education Benelux.
Smith, H. F. (1999). Subjectivity and Objectivity in Analytic Listening. *Journal of the American Psychoanalytic Association , 47* (2), 465-484.
Stone, A. (2004, October 1). eBay Wins a Convert. *Business Week* .
Stone, B. (2008, July 14). Buy.com Deal With eBay Angers Sellers. *The New York Times* .
Stuart, D. (1995). Reputational Rankings: Background and Development. *New Directions For Institutional Research , 88*.
Sundaresan, N., & Moussa, R. (2001). Algorithms and programming models for efficient representation of XML for Internet applications. *WWW '01: Proceedings of the 10th international conference on World Wide Web* (pp. 366-375). New York, NY, USA: ACM.
The Economist. (2004, May 13). At the drop of a hammer. *The Economist print edition* .
Toman, K., & Mlynková, I. (2006). XML Data - The Current State of Affairs. *Proceedings of XML Prague '06 conference*, (pp. 87-102). Prague, Czech Republic.
TreeTagger 3.2. (n.d.). Retrieved from TreeTagger 3.2: http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

Trochim, W. M. (2002). *Trail following of termites.* Retrieved 02 10, 2010, from BIO 208L: http://www.sbs.utexas.edu/jcabbott/courses/bio208web/lectures/scimethod/trail_following_of_termites.htm

Trochim, W. (2002). *Trail following of termites.* Retrieved 02 10, 2010, from BIO 28L: http://www.sbs.utexas.edu/jcabbott/courses/bio208web/lectures/scimethod/trail_following_of_termites.htm

Truijens, O., & Huizing, A. (2005). The Strategic Potential of Information Imperfections: An Information. Strategy Framework for Seeking InfoRent. *PrimaVera Working Paper 2005-14* .

Turney, P. (2001). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. *Proceedings of the 40th Annual Meeting on Association For Computational Linguistics* (pp. 417-424). Philadelphia, Pennsylvania: Annual Meeting of the ACL. Association for Computational Linguistics, Morristown, NJ.

Turney, P., & Littman, M. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Trans. Inf. Syst. , 21* (4), 315-346.

Tweede Kamer: plenaire vergaderingen. (n.d.). Retrieved 06 03, 2009, from Tweede Kamer der Staten Generaal: http://www.tweedekamer.nl/vergaderingen/plenaire_vergaderingen/index.jsp

Van Dale online dictionary: mening . (n.d.). (V. Dale, Producer) Retrieved 06 05, 2009, from Mening: http://www.vandale.nl/vandale/opzoeken/woordenboek/?zoekwoord=mening

Vermeulen, P., & Van der Aa, W. (2003). Organizing Innovation in Services. In J. Tidd, & F. Hull, *Service Innovation. Organisational Responses to Technological Opportunities & Market Imperatives* (pp. 35-53). London: Imperial College Press.

Verschuren, P., & Doorewaard, H. (2007). *Het ontwerpen van een onderzoek* (4e druk ed.). Den Haag: Lemma.

Voss, C., & Zomerdijk, L. (2007). *Innovation in Experiential Services – An Empirical View.* London: London Business School.

W3C. (2008, November 28). *Extensible Markup Language (XML) 1.0 (Fifth Edition)*. Retrieved 06 14, 2010, from http://www.w3.org/TR/REC-xml/

W3C. (2004, 02 10). *World Wide Web Consortium (W3C)*. Retrieved 04 05, 2010, from RDF - Semantic Web Standards: http://www.w3.org/RDF/

Webster, D. (1992). Reputational rankings of colleges, universities, and individual disciplines and fields of study, from their beginnings to the present. In J. Smart, *Higher Education: Handbook of Theory and Research* (Vol. 8, pp. 234-304). New York: Agathon Press.

Weiser, M. (1999). The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev. , 3* (3), 3-11.

Weiser, M., & Brown, J. S. (1997). *The coming age of calm technolgy. Beyond calculation: the next fifty years.* New York, NY, USA: Copernicus.

WekaWiki: Primer. (n.d.). Retrieved 06 10, 2009, from Weka---Machine Learning Software in Java: http://weka.wiki.sourceforge.net/Primer

WekaWiki: Text categorization with Weka. (n.d.). Retrieved 06 08, 2009, from Weka---Machine Learning Software in Java: http://weka.wiki.sourceforge.net/Text+categorization+with+Weka

Wiebe, J. M., Bruce, R. F., & O'Hara, T. P. (1999). Development and use of a gold-standard data set for subjectivity classifications. *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics* (pp. 246--253). College Park, Maryland: Association for Computational Linguistics.

Wiebe, J., & Riloff, E. (2005). Creating Subjective and Objective Sentence Classifiers from Unannotated Texts. In *Computational Linguistics and Intelligent Text Processing* (Vol. 3406/2005, pp. 486-497). Heidelberg: Springer Berlin.

Wigand, R., Picot, A., & Reichwald, R. (1997). *Information, Organization and Management - Expanding Markets and Corporate Boundaries.* Chichester: John Wiley & Sons Ltd.

Wilson, M. (1998). To dissect a frog or design an elephant: teaching digital information literacy through the library gateway. *Inspel , 32* (3), 189-195.

Wilson, T., Pierce, D., & Wiebe, J. (2003). Identifying opinionated sentences. *Proceedings of the 2003 Conference of the North American Chapter of the Association For Computational Linguistics on Human Language Technology: Demonstrations. 4*, pp. 33-34. Edmonton, Canada: North American Chapter Of The Association For Computational Linguistics. Association for Computational Linguistics.

Witten, I., & Frank, E. (1999). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann.

Wright, B. (1991). How the Media Affect College Admissions. *Education Digest , 57* (3).

Yahoo Developer Network. (2010). *Supported Regions in Yahoo! Search Web Services*. Retrieved 06 13, 2010, from http://developer.yahoo.com/search/regions.html

Yin, R. (2003). *Case study research: design and methods* (3rd ed. ed.). London, UK: Sage Publications.

Yu, H., & Hatzivassiloglou, V. (2003). Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. *Proceedings of the 2003 conference on Empirical methods in natural language processing*, (pp. 129-136).

APPENDIX 1 – ENCODINGS FOUND

Header encoding			Pseudo Attribute Encoding			Metadata Encoding		
Encoding	Count	%	Encoding	Count	%	Encoding	Count	%
NULL	137971	76.4%	utf-8	99939	55.3%	NULL	164490	91.1%
utf-8	33879	18.8%	NULL	46657	25.8%	utf-8	8307	4,6%
iso-8859-1	6944	3.8%	iso-8859-1	25237	14.0%	iso-8859-1	6757	3.7%
windows-1251	1044	0.6%	windows-1251	4986	2.8%	windows-1251	418	0.2%
iso-8859-15	428	0.2%	windows-1252	1569	0.9%	iso-8859-15	173	0.1%
windows-1252	177	0.1%	iso-8859-15	795	0.4%	ascii	146	0.1%
iso-8859-2	80	0.0%	ascii	369	0.2%	windows-1252	121	0.1%
koi8-r	77	0.0%	iso-8859-2	261	0.1%	iso-8859-2	83	0.0%
ascii	9	0.0%	windows-874	214	0.1%	utf-16	75	0.0%
windows-1250	6	0.0%	gb-2312	139	0.1%	koi8-r	31	0.0%
iso-8859-9	5	0.0%	tis-620	121	0.1%	shift_jis	12	0.0%
gb-2312	5	0.0%	windows-1250	87	0.0%	windows-1250	8	0.0%
shift_jis	4	0.0%	shift_jis	67	0.0%	windows-874	7	0.0%
windows-1256	3	0.0%	koi8-r	58	0.0%	windows-1256	6	0.0%
gbk	3	0.0%	windows-1256	36	0.0%	gb-2312	2	0.0%
big-5	1	0.0%	utf-16	33	0.0%	euc-jp	1	0.0%
euc-kr	1	0.0%	big-5	15	0.0%	tis-620	1	0.0%
windows-874	1	0.0%	iso-8859-9	14	0.0%	gbk	1	0.0%
euc-jp	1	0.0%	gbk	12	0.0%	euc-kr	1	0.0%
iso-2022-jp	1	0.0%	euc-kr	9	0.0%			
			euc-jp	7	0.0%			
			iso-8859-7	4	0.0%			
			iso-8859-11	2	0.0%			
			windows-1255	2	0.0%			
			utf-7	2	0.0%			
			iso-8859-4	1	0.0%			
			iso-8859-10	1	0.0%			
			iso-3166-1	1	0.0%			
			iso-8859-3	1	0.0%			
			ibm850	1	0.0%			
Total	180640	100.0%	Total	180640	100.0%	Total	180640	100.0%

APPENDIX 2 – ENCODING DETECTION METHODS

Encoding	Used evaluation method	Amount correctly specified	Amount faultly specified
big-5	mb_detect_encoding	16 (100.0%)	0 (0.0%)
euc-jp	mb_detect_encoding	9 (100.0%)	0 (0.0%)
euc-kr	mb_detect_encoding	11 (100.0%)	0 (0.0%)
gbk	mb_detect_encoding	0 (0.0%)	16 (100.0%)
iso-2022-jp	mb_detect_encoding	1 (100.0%)	0 (0.0%)
iso-8859-1	mb_detect_encoding	38938 (100.0%)	0 (0.0%)
iso-8859-10	mb_detect_encoding	1 (100.0%)	0 (0.0%)
iso-8859-11	mb_detect_encoding	0 (0.0%)	2 (100.0%)
iso-8859-15	mb_detect_encoding	1396 (100.0%)	0 (0.0%)
iso-8859-2	mb_detect_encoding	424 (100.0%)	0 (0.0%)
iso-8859-3	mb_detect_encoding	1 (100.0%)	0 (0.0%)
iso-8859-4	mb_detect_encoding	1 (100.0%)	0 (0.0%)
iso-8859-7	mb_detect_encoding	4 (100.0%)	0 (0.0%)
iso-8859-9	mb_detect_encoding	19 (100.0%)	0 (0.0%)
koi8-r	Enca	142 (85.5%)	24 (14.5%)
shift_jis	mb_detect_encoding	0 (0.0%)	83 (100.0%)
utf-7	mb_detect_encoding	2 (100.0%)	0 (0.0%)
utf-8	mb_detect_encoding[15]	142124 (100.0%)	0 (0.0%)
windows-1251	Enca[16]	6062 (94.0%)	386 (6.0%)
windows-1252	entities comparison	1356 (72.6%)	511 (27.4%)

[15] PHP: mb_detect_encoding – Manual: <http://php.net/manual/en/function.mb-detect-encoding.php>
[16] enca(1): detect/convert encoding of text files - Linux man page: <http://linux.die.net/man/1/enca>

APPENDIX 3 – XMLLINT ERROR DOMAINS

Error domain	Description	Error level output in original libxml2 (when different)
parser		
namespace		
XML_FROM_DTD		(no output)
validity		
HTML parser		
memory		
output		
I/O		
Xinclude		
Xpath		
parser		
regexp		
module		
Schemas validity		
Schemas parser		
Relax-NG parser		
Relax-NG validity		
Catalog		
C14N		
XSLT		
encoding		

APPENDIX 4 – OVERVIEW OF LIBXML2 ERROR CATEGORIES

Error Category	Error Domain	Error Level
Input is not proper UTF-8, indicate encoding !	parser	fatal error
Start tag expected, '<' not found	parser	fatal error
Opening and ending tag mismatch	parser	fatal error
xmlParseEntityRef: no name	parser	fatal error
EntityRef: expecting ';'	parser	fatal error
AttValue: " or ' expected	parser	fatal error
attributes construct error	parser	fatal error
Couldn't find end of Start Tag	parser	fatal error
Premature end of data in tag	parser	fatal error
Space required after the Public Identifier	parser	fatal error
SystemLiteral " or ' expected	parser	fatal error
SYSTEM or PUBLIC, the URI is missing	parser	fatal error
Specification mandate value for attribute	parser	fatal error
error parsing attribute name	parser	fatal error
expected '>'	parser	fatal error
Entity not defined	parser	fatal error
StartTag: invalid element name	parser	fatal error
Attribute redefined	parser	fatal error
Extra content at the end of the document	parser	fatal error
Document is empty	parser	fatal error
PCDATA invalid Char value	parser	fatal error
CData section not finished	parser	fatal error
Unescaped '<' not allowed in attributes values	parser	fatal error
Malformed declaration expecting version	parser	fatal error
Blank needed here	parser	fatal error
parsing XML declaration: '?>' expected	parser	fatal error
XML declaration allowed only at the start of the document	parser	fatal error
Comment not terminated	parser	fatal error
Excessive depth in document	parser	fatal error
ParsePI: PI never end ...	parser	fatal error
Comment must not contain '--' (double-hyphen)	parser	fatal error
Sequence ']]>' not allowed in content	parser	fatal error
internal error	parser	fatal error
Char out of allowed range	parser	fatal error
Unregistered error message	parser	fatal error
CharRef: invalid decimal value	parser	fatal error
xmlParseCharRef: invalid xmlChar value	parser	fatal error

xmlParsePI : no target name	parser	fatal error
PEReference in prolog	parser	fatal error
Document labelled UTF-16 but has UTF-8 content	parser	fatal error
Entity failed to parse	parser	fatal error
DOCTYPE improperly terminated	parser	fatal error
Invalid XML encoding name	parser	fatal error
AttValue: ' expected	parser	fatal error
Unsupported encoding	parser	fatal error
chunk is not well balanced	parser	fatal error
switching encoding: encoder error	parser	fatal error
invalid character in attribute value	parser	fatal error
ParsePI: PI space expected	parser	fatal error
CharRef: invalid hexadecimal value	parser	fatal error
Detected an entity reference loop	parser	fatal error
Space required after 'PUBLIC'	parser	fatal error
String not closed expecting " or '	parser	fatal error
xmlParseComment: invalid xmlChar value	parser	fatal error
--[a]	encoding	fatal error
Entity not defined	parser	recoverable error
encoder error	I/O	recoverable error
Namespace prefix is not defined	namespace	recoverable error
Failed to parse QName	namespace	recoverable error
colon are forbidden from PI names	namespace	recoverable error
xmlns: is not a valid URI	namespace	recoverable error
not a valid URI	namespace	recoverable error
Invalid URI	parser	recoverable error
ID already defined	validity	recoverable error
buffer full	I/O	recoverable error
Redefinition of element	validity	recoverable error
Element has too many ID attributes defined	validity	recoverable error
xmlParsePITarget: invalid name prefix 'xml'	parser	warning
xmlns: URI is not absolute	namespace	warning
Unsupported version	parser	warning
Attribute already defined	validity	warning
PEReference: not found	parser	warning
--[a]	namespace	warning
--[a]	--[a]	--[a]

[a] At some error it occurred that an error category, error domain or error level could not be found when the error occurred.