university of groningen

faculty of mathematics and natural sciences

IDENTIFYING RELATIONSHIPS BETWEEN WEBSITES

Relating websites to people

MARK HOEKSTRA

Software Engineering and Distributed Systems
Faculty of Mathematics and Natural Sciences
University of Groningen

September 25, 2015

SUPERVISORS:
prof. dr. ir. M. Aiello
prof. dr. M. Biehl
M. Homminga

ABSTRACT

Finding ownership relations between websites in the almost 1 billion websites available on the Internet is like searching for a needle in a haystack. Although website owners have the possibility of establishing connections with their other websites, it might not always be desired in order to remain anonymous. Even if these connections are established this is not done following a uniform protocol, making the identification of such relations an intricate task.

Nevertheless, identifying these relations yields a broad range of opportunities. One can utilize this information in law enforcement, commerce or use it for optimizing search engines.

In this work I present my investigation on whether it is possible to identify relations between websites using their characteristics. These characteristics consist of identifiers, which are unique across websites, as well as detectable website technologies (e. g. servers, frameworks) serving as features. Logistic regression is used with the features as input and identifiers at the base of the ground truth. Relations between websites are identified with an accuracy of up to 90%, measured with an F_1 score (i. e. harmonic mean of precision and recall).

We are all now connected by the Internet,
like neurons in a giant brain.

— Stephen Hawking (TODAY, 2014)

ACKNOWLEDGMENTS

I would like to use this opportunity to thank the people who supported me during this project.

I want to thank Web-IQ for providing the resources that were needed for finishing this project.

I am grateful for being part of the Web-IQ team who were always willing to spent time by answering all of my questions. Especially, I want to thank Mathijs Homminga, the CTO of Web-IQ, for being my daily supervisor and keeping me on the right track. I would also like to thank Ben Meijering for being a walking encyclopedia of machine learning related information and more.

Finally, I would like to express my sincere gratitude to my supervisors, prof. dr. ir. M. Aiello and prof. dr. M. Biehl for their support and guidance.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

API	Application Programming Interface
AUC	Area Under the Curve
HDFS	Hadoop Distributed Filesystem
HTML	HyperText Markup Language
IE	Information Extraction
IR	Information Retrieval
k-NN	k-Nearest Neighbors
NLP	Natural Language Processing
PII	Personally Identifiable Information
RE	Relation Extraction
ROC	Receiver Operating Characteristic
SEO	Search Engine Optimization
SVM	Support Vector Machine
URL	Uniform Resource Locator

GLOSSARY

evidence	Characteristic that can be extracted from a website, which is useful for identifying relations. 22, 24, 25, 31, 57
fuzzy evidence	Evidence that, when considered on its own, is not valuable for identifying relations between websites, but is when it is considered in combination with other fuzzy evidence. x, 22–25, 27, 31, 59, 61, 71
hard evidence	Evidence which gives an almost hundred percent guarantee that there exists a relation between websites (i. e. a globally unique identifier). 22–25, 27, 31, 44, 59, 61, 65
webmaster	Person or company owning a website or capable of editing a website or both. 2, 14, 16, 18–25, 27–31, 40, 54–56, 59–61

1 INTRODUCTION

Since the advent of the Internet it proliferated immensely: 40% of the world population has access to almost 1 billion websites in 2015 (Stats, 2015). Internet users may freely add content to and remove content from the Internet, creating a complex website topology which is essentially unstructured and unmanaged.

The unstructured and unmanaged nature of the Internet makes it the users' responsibility to specify their identity and relationships to other parts of the Internet. However, this is not required and may not be desired. Not specifying identities and relations to other websites enhances the level of anonymity of an Internet user.

Anonymity on the Internet is pursued, among others, for the sake of preventing misuse, protection against hacking, for acquiring alternative identities and as a means to enhance privacy (Suler, 2004; Rainie et al., 2013; Christopherson, 2007). Besides being used for good intentions, anonymity is used in a great deal of criminal activities.

New types of criminal activities rose since the introduction of the Internet and the Internet provided means to enhance the "success" of existing criminal activities. Criminals developed business models around spamming, phishing, denial of service attacks, malware, adware and tons of other activities. Besides these activities, the Internet greatly simplifies getting access to black markets and child sexual abuse.

To be able to capture criminals, decrease anonymity and provide more structure to the Internet, I examine similarities between the vast collection of websites. These similarities can be used as building blocks for identifying relations between websites. Examples of these building blocks are website technologies and a website's structure, but also identifiers used in the source code of a website.

The main focus in this research is on identifying relations between websites, which involve the people or companies that are responsible for websites. Such a relation may be that multiple websites are owned by the same person or company.

1.1 PROBLEM STATEMENT

Interconnectivity between websites[1] can be defined in multiple ways. Relations between websites can be based on ownership or based on

1 A website is referred to by its homepage (e. g. `www.google.com`) and consists of all pages of a domain.

the geographical location of the area of operation for instance. These relations are not always apparent or detected automatically.

An owner of a website can take the responsibility and list other websites in possession, but this is not required and not always desired.

A great deal of information is lost due to unknown relations. Identifying these relations can help Internet users with, for example, quickly gaining an overview of other websites in possession of a person or company. One can also discover more information about a person by browsing that person's other websites. Another use case can be law enforcement, such as identifying the owner of an illegal website by considering other related websites.

1.2 RESEARCH FOCUS

This research identifies unknown relations between websites. In particular the relations that are considered are relations involving webmasters. For the purpose of this research a webmaster is defined as the person or company owning a website or capable of editing a website or both.

Next to this type of relation, which is described in more detail in Section 4.1.1, research is done on other relations as described in Chapter 3 and Section 4.1.2. Examples of such relations are relations based on websites' functionality or hyperlinks.

To identify these relations in the vast collection of websites, the main question to be answered, which forms the basis of this research, is:

MRQ How can unknown relations between websites be identified?

Ultimately, MRQ is answered by combining the answers to these research questions:

RQ1 In which ways can interconnectivity between websites be defined?

RQ2 What kind of evidence can be used for identifying relations between websites?

RQ3 Can evidence be combined to identify relations between websites and how?

RQ4 How can the results from this research be evaluated?

1.3 RESEARCH PROCESS

To be able to answer the main research question, this research is divided in to four phases where each phase answers a research question.

The first phase of the research is devoted to defining the different types of relations that can exist between websites and provide an answer to the first research question (RQ1).

Phase two concerns the identification and extraction of characteristics from websites in order to answer RQ2.

An answer to RQ3 is given in phase three in which research is done in to combining the extracted characteristics from phase two.

The last phase is dedicated to the evaluation of the results in the previous phases to be able to give an answer to RQ4.

1.4 METHODOLOGY

The description of the methodology is based on the definitions written by Kothari in *Research Methodology, Methods and Techniques* (Kothari, 2011). Kothari defines four research aspects, where each aspect comprises two counterparts.

The first aspect mentioned is whether the research is descriptive or analytical. This research is analytical as datasets are used for the evaluation of the theories.

A solution must be found for a practical problem which makes this research applied instead of fundamental.

The next aspect considers the opposition of quantitative and qualitative research. The research is qualitative as theories need to be established for the detection of relationships between websites.

The last aspect mentioned, is the choice between performing a conceptual research and an empirical research. This research is empirical as hypotheses are tested against datasets and altered if they do not lead to desired conclusions (i. e. the relations identified are incorrect).

1.5 OVERVIEW

The rest of this thesis is organized as follows.

Chapter 2 gives background information which is needed for the reader to comprehend the rest of this work.

An overview of related work is given in Chapter 3 along with a comparison to this work.

Chapter 4 outlines the concept and provides answers to the first two research questions (RQ1 and RQ2).

Chapter 5 is dedicated to the realization.

The evaluation and results are presented in Chapter 6 in which the last two research questions are answered (RQ3 and RQ4).

Finally, Chapter 7 concludes and summarizes this research with an answer to the main research question (MRQ).

2 BACKGROUND

Before describing other advances in the research area, I give background information on assorted relevant aspects.

Data mining, which comprises various fields of research, is covered in Section 2.1. Another area of research involves the crawling of websites as elaborated on in Section 2.2. Big data is considered as well in Section 2.3, which concerns the data produced by web crawling. Lastly, background is given on the field of machine learning in Section 2.4.

These areas of research are described to give background which enables the reader to better understand the rest of this work.

2.1 DATA MINING

Data mining, a step in the process of discovering knowledge from data, is defined as the procedure of extracting patterns from data (Fayyad et al., 1996). Data mining comprises several areas, as defined as follows.

Information Retrieval The field of Information Retrieval (IR) is dedicated to the extraction of relevant unstructured data from a collection of data that satisfies a query (Baeza-Yates et al., 1999; Manning et al., 2008; Salton, 1989) while keeping the number of irrelevant results to a minimum. IR can be applied on a previously created index or by linearly searching through text (i. e. grepping) (Manning et al., 2008). In order to assess the relevance of query results, the performance measures recall and precision can be used as described in Section 2.4.3. An example of increasing the performance of IR techniques for the web considers the structure of a website in order to assess relevance (Cutler et al., 1999; Kim and Zhang, 2001). IR techniques are applied in search engines for the web (Baeza-Yates et al., 1999; Cutler et al., 1999) on a large scale or on a smaller scale in libraries.

Information Extraction Where the field of IR is interested in extracting relevant documents from a collection of documents, Information Extraction (IE) is interested in the structure of a document, hence complementing IR. The area of IE is dedicated to automating the extraction of structured data from raw, and thus unstructured, data (Chang et al., 2006; Gaizauskas and Wilks, 1998; Freitag and McCallum, 1999; Cowie and Lehnert, 1996). Because of this, IE is closely

related to the field of Natural Language Processing (NLP) (Lima et al., 2013), which focuses on the interaction between machines and natural languages (Chowdhury, 2003). Several approaches to IE exist, such as regular expressions and classifiers (Lima et al., 2013). IE is used in various applications, such as indexing news stories (Surdeanu et al., 2003) or extracting locations out of messages (Freitag and McCallum, 1999).

Relation Extraction Relation Extraction (RE) is considered as a part of IE (Culotta and Sorensen, 2004). RE is a field of research aimed at defining relations between entities in texts (Brin, 1999; Bollegala et al., 2009; Culotta and Sorensen, 2004; Chu-Carroll et al., 2012). Bollegala et al. define two categories of similarity measures between relations. One category is the *attributional* similarity measure which measures similarity between words by using their attributes (Bollegala et al., 2009). The other category is the *relational* similarity measure which is about the resemblance of relations between pairs of words (Bollegala et al., 2009). RE is used for finding relations, similar to a relation between a known pair of entities (e. g. acquisition relations) (Bollegala et al., 2009; Chu-Carroll et al., 2012).

website classification Website classification or web page classification is the field of research that is considered prior to using IR, IE and RE techniques (Yu et al., 2004). In general, website classification is the process of assigning categories to websites (Qi and Davison, 2009). Classification can be done in multiple ways (Qi and Davison, 2009), such as basing it on, functionality (Lindemann and Littig, 2006, 2007), genre (Zu Eissen and Stein, 2004; Boese, 2005) or subject (Qi and Davison, 2009). The main application of website classification is, among others, in improving search engines and it is an essential aspect in focused crawling (Qi and Davison, 2009).

Focused web crawling is crawling that is specialized in a certain topic (Novak, 2004)

2.2 WEB CRAWLING

Web crawling is the process of collecting data from the web, and storing it in a database, by queuing hyperlinks encountered for future crawling (Novak, 2004; Olston and Najork, 2010; Faheem, 2012). This process is repeated until no crawling is possible anymore, or when a certain boundary is exceeded.

Commonly, crawling is initiated from a set of seed pages, which provide the initial hyperlinks for the crawling process (Novak, 2004). The seed pages indicate the kind of content needed for the goal of the crawling. The IR, IE and RE techniques, outlined in Section 2.1, can be used on the crawled data for processing the information. The website classification techniques from Section 2.1 allow for focused crawling on websites with specific functionalities or topics.

For crawling social media, websites, such as Twitter and Facebook, are useful seeds

An important application of web crawling is in search engines. Search engines allow users to pose queries against websites which are previously crawled and indexed (Olston and Najork, 2010). Other uses of web crawling are web data mining and web monitoring (Olston and Najork, 2010). Another application of crawling is archiving the social web to answer social and current questions (Faheem, 2012).

2.3 BIG DATA

In 2009, Jacobs defined big data as "data whose size forces us to look beyond the tried and true methods that are prevalent at that time" (Jacobs, 2009). A similar definition is given by Chen et al. who define big data as "...datasets that could not be perceived, acquired, managed, and processed by traditional IT and software/hardware tools within a tolerable time" (Chen et al., 2014). By this Jacobs and Chen et al. want to capture the change of the definition over time, because what we consider big data now, may not be so "big" in the future.

1TB of data used to be considered as a tremendous amount of information

Web crawling is one of the sources of big data and conventional databases, such as relational databases, cannot efficiently cope with the amount of crawled data. According to Jacobs we have to look beyond the "tried and true methods" and this is exactly what has happened.

Other big data sources include data from sensors (e.g. satellites) and machine log data

The invention of NoSQL databases, such as MongoDB, CouchDB and HBase, which allow for non-relational storage instead of relational storage, generally provide better performance, flexibility and scalability for large amounts of unstructured data than conventional databases (Stonebraker, 2010; Hecht and Jablonski, 2011; Leavitt, 2010).

Information in big data, which can be utilized by data mining, is used for various purposes, such as analysis or predictions. An example of analysis and prediction on big data is the early warning system for flu designed by Google (Helft, 2008; Butler, 2008). The system is able to significantly reduce the detection time for flu outbreaks, by predicting the spread of the flu, through "...extracting patterns of flu-related search terms ..." (Butler, 2008). Another usage is analysis performed on data gathered from telescopes and information gathered from consumer transactions of retail corporations (Cukier, 2010).

2.4 MACHINE LEARNING

The definitions included in this Section are used as defined in *Introduction to Machine Learning* by Alpaydin (Alpaydin, 2010).

For some computer challenges, algorithms cannot simply be devised to obtain the desired output from the input data. An example of such a challenge is predicting the gender of a person based on fea-

tures such as length, weight, shoe size, etc. For these challenges, such as the example and similar, the issue is not a lack of data, but rather our understanding of the data or our incapability of detecting trends in the data.

Given enough data and features, a process can be created which is able to detect patterns in the data. This process is approximate and is called machine learning. Machine learning is used in a wide range of applications, such as: detecting consumer patterns, classification of diseases and in this research, identifying relations between websites.

The pipeline of a machine learning process generally consists of three steps:

- data collection and preparation;
- application of one (or more) machine learning algorithm; and
- verification of results.

2.4.1 *Data Collection and Preparation*

As seen in the previous Itemization, the first step in machine learning is to collect data. Data collection may be done using a survey, gathering information through sensors or crawling the Internet. Features are identified, which are the pieces of information in the data and represent the characteristics of entities. Examples of features, from data detected in a survey, are gender and age. Humidity and temperature are possible features measured by sensors. Features that can be obtained through web crawling are ip addresses or a website's ten most used words.

Besides collecting the data, the data must be prepared in a format suitable for machine learning. This preparation involves creating two separate datasets. One set serves as a collection of features, also named independent variables, which forms the input. An example is a set of features for people, where a feature can be a person's length or weight. The other dataset is a set which contains, for each entity, the entity's class and is called the target set which forms the dependent variables and serves as the output. The class is that what needs to be predicted and could be the gender for a person. The target set must be labeled in advance and is also called the ground truth.

2.4.2 *Machine Learning Algorithms*

Once the data is collected and prepared it can serve as the input for different types of machine learning algorithms. A machine learning algorithm can be divided in to the following three types, depending on the learning method: supervised, unsupervised or reinforcement learning.

Supervised learning algorithms are algorithms which learn output from a certain input, where the output is determined by a supervisor.

Unsupervised learning on the other hand is a form of machine learning which only considers the input. Therefore, there is no supervisor labeling the data.

Lastly, reinforcement learning is the category in which the output is a sequence of actions. An example of an application, of algorithms in this category, is game playing in which a single step is not important whereas a sequence of steps is.

Maze solving is an example of reinforcement learning, where a step towards the exit is considered a good step

Another distinction of machine learning algorithms can be made by considering their output. Classification, regression and clustering are types of algorithms which focus on different types of output.

In classification, the problem is that of predicting the class from the input, such as detecting a person's gender.

Regression is the problem of predicting a number from the input, such as predicting a person's weight based on gender and length information.

Clustering is another type of problem in which the output is produced by labeling the input in to groups which are unknown in advance.

2.4.3 *Verification of Results*

The last step in applying machine learning is verifying the results. This Section covers different approaches for verifying machine learning results.

For a two-class problem there are four possible cases of classification as shown in Table 1. True Positive (TP), in the upper left corner, is a case where the outcome is correctly predicted as positive. True Negatives (TN) are classes which are correctly predicted as negative. False Positive (FP) classes are ones that are falsely predicted as positive. Lastly, the upper right corner contains False Negative (FN) classes which specify classes that are falsely predicted as negative.

From Table 1 it can be seen that a distinction between two types of errors is made. One type of error is FP, falsely labeling a positive case, and the other type of error is FN, falsely labeling of negative case. A diversity of performance measures exists which use the cases in the confusion matrix as a base for the verification of the machine learning results.

One such a performance measure is Receiver Operating Characteristic (ROC). ROC is a graphical plot which shows a curve which rep-

Table 1: Confusion Matrix of a Two-Class Problem

		Prediction outcome		
		p	n	*total*
actual value	p'	True Positive	False Negative	P'
	n'	False Positive	True Negative	N'
	total	P	N	

resents the TP-rate against the FP-rate for certain thresholds. The TP- and FP- rate are respectively defined in Equation 1 and Equation 2.

$$\text{TP-rate} = \frac{TP}{P'} \tag{1}$$

$$\text{FP-rate} = \frac{FP}{N'} \tag{2}$$

In words, the TP-rate in Equation 1 can be defined as the share of correctly labeled cases. The FP-rate in Equation 2 can be defined as the share of classes falsely labeled as positive.

The thresholds for the ROC plot are θ values which represent a condition. If the probability of the positive class is higher than a threshold close to 1, a few to no classes are labeled as false positives, but also a few are labeled as true positives. Decreasing the threshold increases the number of true positives while increasing the chance of introducing false positives.

An ROC curve can be created by selecting different thresholds to generate a selection of TP- and FP-rate values as shown in red in Figure 1. A classifier with a TP-rate closer to one and an FP-rate closer to zero moves the curve to the upper left corner. The diagonal represents a classifier which produces as many correct decisions as incorrect decisions. Any classifier along this diagonal is the worst classifier that can exist as any classifier towards the lower right corner can swap its incorrect and correct decisions. The ROC curve is a visual analysis and can be reduced to one number by calculating the AUC.

An Area Under the Curve (AUC) of 1 represents an ideal classifier

In IR, queries are performed against a dataset. These queries return subsets of the dataset (or the dataset as a whole) as results. To verify the results of such queries, the relevance of these subsets to the

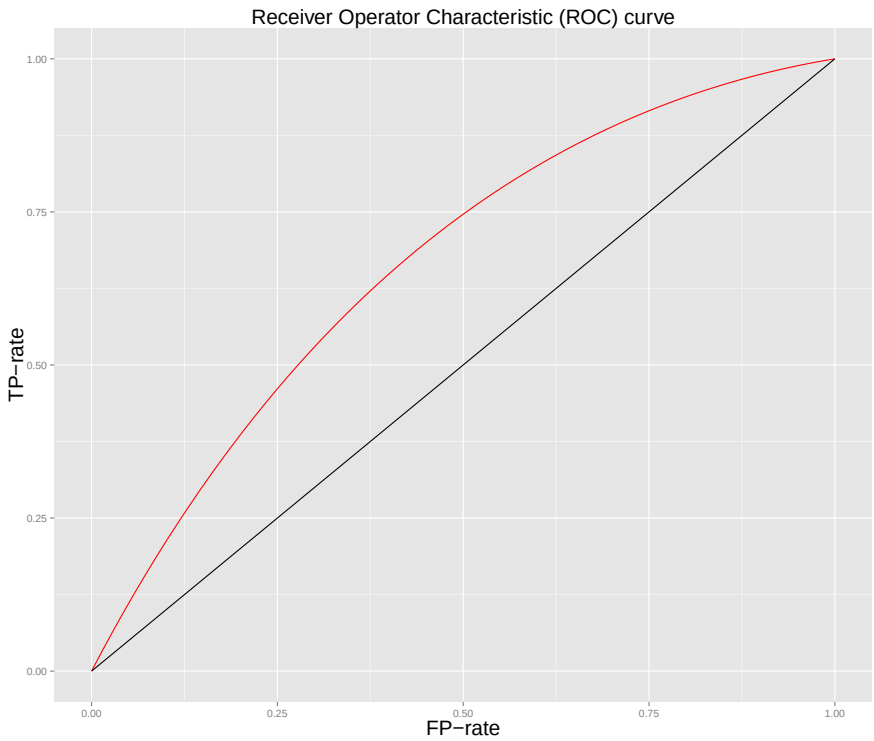


Figure 1: Plot of a ROC curve

query must be assessed. Assessing the relevance of the results alone, however, does not take the rest of the dataset in to account. Therefore, besides assessing the results, the data not returned as a result must be evaluated.

The process of assessing the relevance of the results is called precision or accuracy. Thus precision considers the share of the results that are relevant to the query. The results that are relevant to the query are true positives (TP), while the results that are not relevant are false positives (FP). The results of a query are the sum of TP and FP which is indicated by P as seen in Table 1. With these interpretations, precision is defined as:

$$\frac{TP}{P} \tag{3}$$

Since precision only considers data returned as an answer, another metric is needed which considers the data not returned as an answer. This other metric is called recall. Recall measures the data that should have been returned as an answer but is not. Recall, therefore, considers P' which is the sum of TP and FN and is defined as:

$$\frac{TP}{P'} = \text{TP-rate} \tag{4}$$

Precision and recall can be combined in to one performance measure, called F_1 score. F_1 score is defined as:

$$2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{5}$$

The F_1 score is a harmonic mean and is used to reason about the machine learning results by representing recall and precision in one value.

In order to assess the results, the fitted machine learning model is usually tested on a dataset other than the dataset used for fitting the model. This process is called cross validation and is realized by splitting the dataset in two sets: the train set used for fitting and the test used used for verification.

2.5 SUMMARY

Data mining techniques are used to gain knowledge from raw data. The relevant key aspects of data mining are Information Retrieval (IR), Information Extraction (IE), Relation Extraction (RE) and website classification.

Data mining is performed on collected data. Web crawling is a means to collect data and concerns storing webpages by following hyperlinks.

Big data is the name for large collections of data which can be obtained from different sources, such as from sensors and by web crawling. One of the challenges of big data is to process a large amount of data in a reasonable amount of time.

Machine learning is the process of detecting trends in a dataset without having to devise an algorithm. Machine learning techniques use data with known output in order to predict the output on data not previously encountered. The machine learning process starts with the collection and preparation of data followed by the application of machine learning algorithms. Subsequently, the results are evaluated, which can be done with several performance measures.

3 RELATED WORK

I discuss work related to this research concerning relations between websites in this Chapter.

Section 3.1 depicts the research about authors of text. Websites related to each other through similarity are considered in Section 3.2. Research on website relations based on hyperlinks is described in Section 3.3. Websites can also be related to each other by their users as shown in Section 3.4. Section 3.5 considers the work on website relations using identifiers. Finally, Section 3.6 gives an overview of the related work.

3.1 IDENTICAL AUTHORS

Extensive research is done on the field of authorship identification (also known as authorship attribution) which is part of a broader research area called authorship analysis. Authorship identification concerns the differentiation between texts written by different authors (Stamatatos, 2009).

The task of identifying authors has planted its roots in a large number of applications (Madigan et al., 2005). Examples, given by Madigan et al., of such applications are relating certain texts to known criminals, identifying the authors of copyright protected material and discovering authors of computer viruses.

Authorship identification uses a set of candidate authors for comparison (Stamatatos, 2009). Finding two pieces of text from one author in a large dataset containing different types of texts with a huge number of different, possibly anonymous, authors increases the difficulty of applying authorship identification for finding related websites (Abbasi and Chen, 2005).

Authorship identification mainly uses machine learning, Information Retrieval (IR) and Natural Language Processing (NLP) techniques (Stamatatos, 2009). Machine learning proves to be useful, because authorship identification of texts from unknown authors is based on a training set, containing texts related to known authors (Koppel et al., 2009). IR techniques contribute to the challenge of authorship identification as it allows for the classification of texts (Koppel et al., 2006; Stamatatos, 2009). Finally, research in NLP produced new methods of performing analysis on texts and resulted in the capability of extracting new features regarding style of texts (Stamatatos, 2009).

Additionally, authorship analysis focuses on verifying if a text is written by a specific author and detecting plagiarism (Stamatatos,

2009). Profiling authors from texts (i. e. extracting distinctive information, such as age, sex, etc.) and detecting inconsistencies in texts are other aspects of authorship analysis (Stamatatos, 2009).

This research does not focus on authorship identification as this is an entirely different area of research and beyond this research's scope. On the other hand I am interested in relating websites to each other from the same author in certain cases. For this, features, such as author information in HyperText Markup Language (HTML) meta tags or copyright information, can be extracted and provide information regarding authors.

3.2 SIMILAR WEBSITES

Relations between websites can be defined based on their similarity. Similarity, in this sense, is defined as having the same functionality or content or both. Finding similar websites is not the main focus of this research, but similar websites might be encountered when looking for relations involving webmasters.

Numerous search engines are developed around this principle. SimilarSites[1], a product of SimilarGroup[2], is such a search engine and is able to find websites similar to a specific website. According to the developers, several techniques, for finding similar websites, are applied on previously crawled data, such as website classification, analysis on a website's content and hyperlink structures. Moreover, SimilarSites uses information about other websites visited by Internet users and ratings and reviews from users of SimilarSites.

SimilarSites is one of the many search engines that can be used for finding similar websites. One other search engine worth mentioning is Google. Google allows users to search using the search term "related:". Using this keyword in front of a Uniform Resource Locator (URL), allows the user to find websites similar to the website of that URL.

3.3 WEBSITES REFERENCES

The study of Park et al. is aimed at building a network of affiliates from the most popular websites according to Korean web users (Park et al., 2002). A website is considered to be an affiliate when this website is linked to in a webpage dedicated to affiliates. Park et al. discovered that banking and stock websites are affiliates to a large number of websites. They conclude from this fact that these kinds of websites are crucial in commercial websites.

References between websites (i. e. hyperlinks) can be used as indicators of relations between them. The meaning of a relation based

1 SimilarSites: `http://www.similarsites.com/` (Accessed: March 9, 2015)
2 SimilarGroup: `http://www.similargroup.com/` (Accessed: March 9, 2015)

on a hyperlink depends on the context. In the study of Park et al., a reference between websites identifies an affiliate relationship.

A number of tools use the information of hyperlinks between websites to identify different kinds of relationships. An example of a relationship, that can be identified by interpreting hyperlinks, is the relationship between first and third parties. The Firefox add-on Lightbeam "... visualizes the relationships between the sites one visits and the third party sites that are active on those pages"[3] and uses this information to make the Internet less opaque.

A third party is any party other than the party with which a user interacts

Other applications include improving the quality of finding similar websites using SimilarSites and the "related:" search term in Google as both described in Section 3.2.

Fundamentally, these kind of relations are not the focus of this research. However, using hyperlinks could be explored in specific scenarios, such as when a list of references to other websites, owned by a person, is given on this person's website.

3.4 INTERNET USERS

Besides hyperlinks as a basis for a relation between websites, websites can be related to each other based on an Internet user's surfing behavior as used in SimilarSites described in Section 3.2.

If a user visits website Y, just after visiting website X, a link can be made between websites X and Y. Such browsing behavior can be tracked using cookies stored on a user's local machine (Krishnamurthy and Wills, 2009).

An example of the utilization of such information is the clustering seen in The Internet Map[4]. The Internet Map is based on statistics from Alexa gathered in 2011[5]. The tool visualizes websites as circles where their size represents a website's traffic. The location of a circle is defined by links between websites. A link is made between two websites if a user switches from one website to another. The more switches from users between websites, the stronger the link between them, thus the greater the proximity of these websites. Several clusters can be identified in The Internet Map, such as websites from one country, social network websites or online market places.

Tracking a user's behavior is one way to identify a relation between websites based on Internet users. Another way is based on websites' target audience. If different websites are targeting the same audience, they are most likely operating in the same branch.

This type of relation is utilized in Google AdWords[6]. Customers, of the advertising service Google AdWords, define keywords which trig-

3 Lightbeam for Firefox: https://www.mozilla.org/en-US/lightbeam/ (Accessed: March 6, 2015)
4 The Internet Map: http://internet-map.net/ (Accessed: March 6, 2015)
5 Alexa Internet: http://www.alexa.com/ (Accessed: March 6, 2015)
6 Google AdWords, https://www.google.com/adwords/ (Accessed: March 9, 2015)

ger advertisements based on Google users' search queries. Additionally, Google AdWords provides a feature[7] which allows for offering advertisements to new visitors, similar to existing site visitors, in order to reach a possibly interested audience not previously addressed. According to Google, such a similar audience is based on behavioral cookie tracking.

Although this research is interested in relations that involve people, the interest lies in the relations involving webmasters and therefore relations regarding Internet users are not considered.

3.5 RELATIONS BASED ON IDENTIFIERS

Whereas the previous Sections describe work with different goals and methods, this Section comprises work with similar goals and methods.

DomEye[8] and SpyOnWeb[9] are examples of search tools which are able to utilize identifiers from Google Analytics and Google AdSense for finding related websites. Additionally, DomEye, is able to use other identifiers which, except for eight, are not documented. Both of these websites are able to extract identifiers from websites and compare these to websites that are previously crawled. This research is different from websites such as DomEye and SpyOnWeb as more and different types of identifiers are considered. Furthermore, this research considers characteristics from websites, which are useful in combinations for identifying relations. It is suspected that such characteristics, although not documented, are used in the big search engines. Google, for example, can use it to find related websites to optimize its search engine.

Other methods, which use identifiers, include reverse WhoIs lookup and reverse IP lookup. Reverse WhoIs lookup exploits information belonging to the domain registrant. This information can be accessed by websites such as WhoIs[10] and can be used to find other domains of a person. Reverse IP lookup is the process of discovering all websites that are hosted on the same server. This process might result in a list of websites owned by one person, but it may also result in a gigantic list of unrelated websites sharing one host.

3.6 SUMMARY

To give an overview of the related work it is summarized in Table 2. Table 2 contains the techniques and aims for each type of work. Addi-

7 Similar audience in Google AdWords: https://support.google.com/adwords/answer/2676774?hl=en (Accessed: March 9, 2015)
8 DomEye: http://xaviesteve.com/domeye/ (Accessed: March 18, 2015)
9 SpyOnWeb: http://spyonweb.com/ (Accessed: March 18, 2015)
10 WhoIs: http://www.whois.com/ (Accessed: March 18, 2015)

tionally, differences are outlined along with a reference to the Section which describes the related work.

Table 2: Summary of Related Work comprising techniques, aims and differences

RELATION TYPE	TECHNIQUE	AIM	DIFFERENCE	SECTION
Authorship identification	Machine learning, IR, NLP	Identify author, plagiarism detection, profile authors, inconsistency detection	Not (necessarily) webmaster related	Section 3.1
Similar websites	Website classification, hyperlink structures, behavioral tracking	Discovering similar sites	Not webmaster related	Section 3.2
Website references	Hyperlink	Identify website relations (e.g. affiliates, third party), improve quality of similar site identification	Not webmaster related	Section 3.3
Internet users	Behavioral tracking	Monitor users' browsing behavior	Not webmaster related	Section 3.4
Identifier based relations	Unique identifiers	Identify website relations	Limited number of identifiers, no characteristics	Section 3.5

4 CONCEPT

The aim of this research is to identify relations between websites involving webmasters as described in Section 1.2. To realize this aim, different types of relations are explored in order to provide an answer to RQ1 in Section 4.1.

To answer the second research question (RQ2) an investigation is performed in to the characteristics that can be used for the identification of relations between websites in Section 4.2.

4.1 INTERPRETATION OF WEBSITE RELATIONS

As this research aims at identifying relations between websites involving webmasters, the first question (RQ1) that needs to be answered is about identifying the sorts of relations that can exist between websites and determining which of these types of relations are of interest. The relation of interest is given in Section 1.2, whereas other kinds of relations are described in Chapter 3.

This Section contains a more detailed description of the relation of interest in Section 4.1.1. Relations not further considered are described in Section 4.1.2.

4.1.1 *Relation of Interest*

We consider one relation that is of interest. Here, I define this relation and provide motivations for it being the main focus.

WEBMASTER RELATION The webmaster relation is an important relation, because it is directly related to people. For the purpose of this research a webmaster is defined as described in this Paragraph.

A webmaster can be the person or company owning a website. The owner of a website is the person or company to whom a website is registered. Such an ownership relation between websites can be of interest to Internet users for the purpose of discovering other websites registered to an owner. Law enforcement might also be interested in this relation as it might help solve cases.

A webmaster can also be the person who is able to edit the contents of a website (e. g. administrator or maintainer). If the person editing a website is not the owner, this person might be of more interest. The use cases for this relationship are similar to the previously described ownership relation. However, the identification of these relations can differ greatly. Where information about an owner can possibly be

extracted from the domain registration, information about the administrator or maintainer cannot. On the other hand, information about the administrator or maintainer can possibly be extracted by looking at information in the footer of a website. Evidently, information about the owner might be found in the footer as well.

Another interpretation of a webmaster considers the author of a website. There is a thin line between the use case for authorship relations and the two previously defined interpretations. However, authorship relations are more associated with the content of a website than the website as a whole. A website can, for example, consist of news articles in which the individual authors are less characteristic for a website than the one person owning a website in which he or she writes blogs. In Section 3.1 this relation is described in more detail and an explanation is given about how the identification of this relation can be achieved. As previously explained in Section 3.1, I do not further focus on authorship identification, because it is a different area of research and out of the scope of this research.

4.1.2 *Other Types of Relations*

Along with the relation described in Section 4.1.1, there are relations between websites which are not of interest. These relations are defined in this Section and are not further considered as they essentially do not involve webmasters. Related work on the relations described in this Section is given in Chapter 3.

Yahoo and Bing are the first results when using the search query "related:google.com" in Google (as of March 10, 2015)

SIMILARITY RELATION Multiple websites can be related to each other based on similarity as described in Section 3.2. Websites are similar if they have the same functionality or the same kind of content. An example of websites, considered to be similar, are search engines `www.google.com`, `www.yahoo.com` and `www.bing.com` as they provide the same functionality.

REFERENCE RELATION In Section 3.3, relations between websites are defined based on hyperlinks. A relation between two websites is identified if one of the two, or both, link to each other. In the example of the previous paragraph, Google is related to Bing and to Yahoo as Google refers to Bing and Yahoo by matter of hyperlinks. This relation can be considered in this research when it defines a relation that can help identify the relation described in Section 4.1.1. An example of when this relation is useful, is when a web page contains an overview of references to other websites owned by the same webmaster. Another example is a footer of a website referring to the company owning the website.

INTERNET USER RELATION Another relationship that can be identified is a relationship involving Internet users as mentioned in Section 3.4. This type of relation can be defined in multiple forms.

Two websites may be related if a user browses from a website to another website. This browsing can be tracked by behavioral tracking cookies stored on a user's computer. These cookies store information about a user's actions and maintain a list of websites previously visited (Krishnamurthy and Wills, 2009). An example of this relation is visiting `www.bing.com` from `www.google.com` in the previous example. If no behavioral tracking cookie is active, this relation is not identified.

Another type of relation based on Internet users involves the target audience of websites. The main purpose of defining this relation is to help identify similar websites. Similar websites often target the same audience. The search engines Google, Yahoo and Bing target people wanting to search the Internet. Two different websites selling used cars target people interested in buying used cars. Surfing behavior of users is extracted from tracking cookies and can be used to build user profiles (Krishnamurthy and Wills, 2009). These user profiles can contain information about the websites visited and search terms entered. From these user profiles, patterns can be extracted which help in finding similar audience for targeted advertising (see Section 3.4).

Both of these relations, although concerning people, are not considered in this research as the focus of this research is on combining websites, not on its visitors, but on its webmasters.

GEOGRAPHICAL RELATION The last type of relationship described in this research is based on the geographical locations of websites. This relationship is defined if the host services of multiple websites are geographically located in the same area. This relation is also defined when the areas of operation of multiple websites are in the same area. The area of operation is defined as the home address of a person or the contact address of company.

This type of relation can offer different kinds of information based on the radius of the area. If the radius of the area is defined as 10 kilometers, all companies inside the area can said to be located in the same city. Increasing the radius can provide information about a province or country.

Fundamentally, this relation does not involve a webmaster. However, it might prove useful for identifying relations involving webmasters. One person could, for instance, host all his or her websites on a single host or a company might own multiple websites registered to a single address.

4.2 EVIDENCE EXTRACTED FROM WEBSITES

Section 4.1 provides an answer to RQ1. The next question to be answered is RQ2 and concerns the issue of which evidence that can be used for identifying relationships. Evidence can be defined as a characteristic that can be extracted from a website which is useful for identifying relations involving webmasters.

This research distinguishes two types of evidence, hard evidence and fuzzy evidence respectively described in Section 4.2.1 and Section 4.2.2.

4.2.1 *Hard Evidence*

Evidence is considered "hard" when there is almost a hundred percent guarantee that websites are related to each other. Evidence, in the hard evidence category, appears in the form of identifiers in websites. These identifiers are unique across websites and placed manually by a webmaster. Since identifiers are unique and placed manually it is assumed that websites containing the same identifiers are related. Investigation in to this assumption is performed in Section 5.1.1.

Hard evidence is grouped in to several categories, which are explained in the next Paragraphs. These categories are identified using personal experience and the categorization from Ghostery. Ghostery is a plugin, available for various web browsers, which is able to block requests, from a website being visited, to third parties[1]. Ghostery aims at giving its users more control over their privacy by revealing and controlling requests made to third parties, just as Lightbeam described in Section 3.3. Besides this goal, Ghostery allows users to opt-in to anonymously share detections of these requests and where they are detected in order to help businesses create advertisements.

ANALYTICS Analytics services, such as Alexa[2] and Google Analytics[3], are services which allow websites to track their visitors (Waisberg and Kaushik, 2009). The analytics services provide code snippets which must be placed in websites for the services to be functional. Contained in these code snippets is an identifier which serves as a connection between the website and the account registered for a certain analytics service. This account gives access to the statistics gathered, enables webmasters to provide payment information when necessary, and contains settings which can be altered. This identifier is of interest as it is considered to be hard evidence. Such an identifier, when encountered on multiple websites, is an indication that

1 Ghostery: https://www.ghostery.com/en/ (Accessed: March 10, 2015)
2 Alexa Internet: http://www.alexa.com/ (Accessed: March 10, 2015)
3 Google: http://www.google.com/analytics/ (Accessed: March 10, 2015)

these websites share a connection through the same analytics account maintained by a webmaster.

ADVERTISING A plethora of advertising services exist, which allow for easy placement of advertisements on websites. Webmasters can register their websites to advertisement services. Similarly to analytics services, a code snippet must be included in a website for the advertisements to become active. Again this code snippet contains a unique identifier which can be used as hard evidence.

BEACON Ghostery defines web beacons as objects which serve no purpose other than tracking. Such beacons exist in various forms. Web beacons, which can appear as invisible 1 by 1 pixel images, are means to send cookie information to third parties (Martin et al., 2003; Sipior et al., 2011). Web beacons also work with code snippets containing identifiers which link the website to an account registered at the service providing the web beacons.

Web beacons can be found by searching for image tags, in the HTML source code, containing links to third parties

WIDGET Widgets are, as defined by Ghostery, objects which provide certain functionalities besides tracking. Thus a web beacon with a functionality falls under the widget category. Widgets containing identifiers can be used as hard evidence, while widgets without identifiers can serve as fuzzy evidence (see Section 4.2.2).

VERIFICATION Verification methods are mechanisms which allow webmasters to prove that they are capable of editing a website. The mechanism works by adding a unique identifier to a website, usually in a meta tag. Other methods of verifying the capability of editing a website exist as well, but they are not of interest for this research as they do not contain unique identifiers. An example is uploading a certain file to the server running the website. These mechanisms are widely applied in webmaster tools, which only show certain information about a website after verifying that one is able to edit the website.

TOOL OR SERVICE Websites contain tools or services which enhance their functionality or add new features. These tools often require users to register before being allowed to use the tools in their websites. Along with the registration, the user receives a unique key which is used in the tool's Application Programming Interface (API) requests. Users might use this key, and thus a certain tool, for multiple websites. Detecting this key on multiple websites can indicate relationships between them. An example of such a tool, which requests the use of a key, is reCAPTCHA[4]. reCAPTCHA is a tool which is capa-

4 Google reCAPTCHA: `http://www.google.com/recaptcha` (Accessed: March 10, 2015)

ble of protecting a website against spam by distinguishing between people and robots.

PERSONALLY IDENTIFIABLE INFORMATION Personally Identifiable Information (PII) can also be considered as hard evidence. PII is defined as "...any information about an individual maintained by an agency, including (1) any information that can be used to distinguish or trace an individual's identity, such as name, Social Security number, date and place of birth, ... ; and (2) any other information that is linked or linkable to an individual, such as medical ... and employment information" Office (2008).

PII, such as email addresses or phone numbers, can be used to identify webmaster relations. If an email address is encountered in the footer of multiple websites, then this can be an indication that these websites are owned by the same person or company. An email address encountered in a list of users and their email addresses might not be an indication of such a relation. This means that the location of PII in a website might be important for how it contributes to identifying relations between websites.

4.2.2 *Fuzzy Evidence*

The name fuzzy evidence is partly derived from fuzzy sets (Zadeh, 1965). Fuzzy sets are sets in which the contents cannot precisely be defined. Zadeh uses the example of all numbers greater than 1 as a fuzzy set.

Fuzzy evidence is regarded as evidence that alone is not valuable for identifying relations between websites, whereas it is more useful in combination. An example of fuzzy evidence is the top ten most used words in a website. On its own this might not be enough to identify a relation between websites, but with another piece of fuzzy evidence, the time of a website's registration for example, the chance of identifying such a relation can become greater.

Ghostery defines one more category, besides the categories discussed in Section 4.2.1. This category does not contain tools or services using identifiers and can therefore not be used as hard evidence. The category can, however, be used as fuzzy evidence. This category, named privacy, is a category which contains privacy notices and other privacy related elements according to Ghostery. The use of a certain privacy notice, alongside other pieces of fuzzy evidence, could increase certainty of correctness of an identified relation.

Tools, in the categories discussed in Section 4.2.1, which do not contain identifiers are not used as hard evidence, but can be used as fuzzy evidence similarly to the contents of the privacy category.

Fuzzy evidence is a broad category containing many different types of evidence. In theory, every piece of information detectable on web-

sites can be considered fuzzy evidence. The type of fuzzy evidence that is considered in this research is technologies, such as programming languages, servers and frameworks.

4.3 SUMMARY

There exist many types of relations between websites. There are relations based on webmasters, which form the focus of this research and there are other types of relations, namely similarities between websites, hyperlinks, geographically based relations and relations based on users' behavior. The investigation in to these different types of relations answers RQ1: in which ways can interconnectivity between websites be defined?

Evidence on websites that can be used to identify relations is separated in two categories: hard evidence and fuzzy evidence. Hard evidence is the category containing identifiers which are unique across websites on the Internet. Identifiers are used, among others, for advertising and analytics services. Evidence in the fuzzy evidence category can be any characteristic that is detectable on websites. Identifiers alone are able to relate websites to each other whereas fuzzy evidence identifies relations by combining the characteristics. RQ2 is answered by listing the evidence that can be used to identify relationships between websites.

5 REALIZATION

With answers to RQ1 and RQ2, the extraction of characteristics from websites is performed as portrayed in Section 5.1. In addition, the impact of encountering a shared identifier between websites on a webmaster relation is investigated in Section 5.1.1.

Furthermore, the process of combining the extracted characteristics using machine learning is described in Section 5.2 along with an overview of the tools used in Section 5.3.

5.1 HARD AND FUZZY EVIDENCE

Hard evidence and fuzzy evidence are extracted from previously crawled websites by matching patterns and string comparisons. An overview of the implemented hard evidence identifiers is given in Appendix A. This Appendix lists, for each identifier, its name, a reference to its website and the code snippet which must be placed on a website in order for the service to function. In total 28 identifiers are detected in 6 categories. Figure 2 shows how the implemented identifier types are spread over the categories.

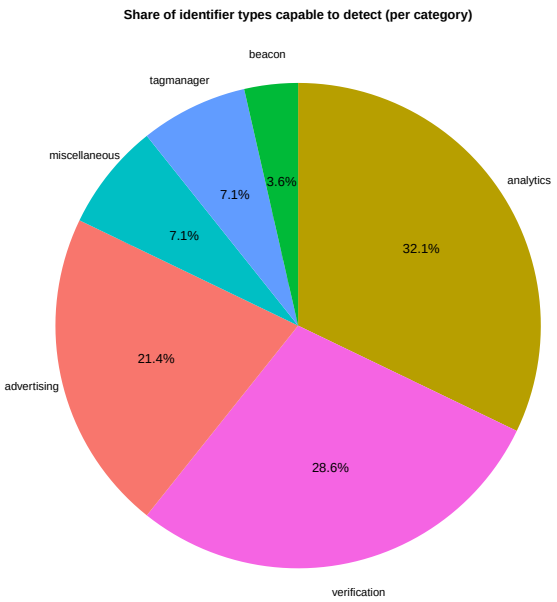


Figure 2: Pie chart of the share of implemented identifier types per category

Appendix B gives a list of 49 features in the fuzzy evidence category which can be detected. For each feature, its name, a reference to its website and a short description is given. The term feature is used

in order to comply with the jargon in machine learning. Features are "... the result of measurements made on an individual or event" (Alpaydin, 2010). Examples of features are the weight and gender of a person or the price of a car. In this research I use website technologies as features, such as frameworks and programming languages.

5.1.1 *Study: Hard Evidence*

In the beginning of Section 4.2.1 an assumption is made about what identifiers can contribute to the identification of webmaster relations between websites. In this Section proof for this assumption is investigated by manual verifying relations between websites based on identifiers. This investigation is performed on websites encountered in the dataset that is used in this work.

This dataset is described in Section 6.1. For the purposes of this investigation it is only necessary to know that the dataset contains around 20.000 websites of which about 28.000 pairs of related websites are identified. A related pair, in this context, means that two websites contain at least one shared identifier.

In this study, a subset of the 28.000 related pairs is considered and the relations between the websites are manually verified. The subset that is considered consists of websites that are related by ten separate identifiers.

Table 3 presents a list of ten identifiers, with the number of websites they are detected on and an explanation of the relation. Horizontal lines between the rows separate different clusters of websites (e.g. the first two rows show two identifiers in which the second identifier occurs on a subset of websites containing the first identifier).

The formula for calculating the number of related pairs that can be created from a cluster of websites is given in Equation 6.

$$f(n) = \frac{n^2 - n}{2} \tag{6}$$

where n is the number of websites that contain a shared identifier.

For each cluster in Table 3, the identifier relating the highest number of websites to each other is investigated (i. e. the biggest cluster of websites). It is observed that the other identifiers generate clusters of websites which are subsets of the biggest cluster. Therefore, the subsets do not add information regarding related websites that is not included already in the biggest cluster.

Applying Equation 6 on the 5 identifiers, with the highest number of websites per cluster, results in 8.195 pairs as can be seen in Equation 7.

$$f(27) + f(7) + f(90) + f(68) + f(56) = 8195 \tag{7}$$

Table 3: Study of relations between websites

IDENTIFIER	#WEBSITES	RELATION
ComScore:6036461	27	TripAdvisor
ua:UA-30198665-1	26	
QuantCast:p-c1rF4kxgLUzNc	7	StackExchange
ComScore:17440561	7	
ua:UA-51861146-1	90	Gatehouse Media
BingWebmasterTools:7E15F9269E2CE66F2A488ABB04B5015E	89	
ua:UA-6842750-1	87	
ShareThis:189e1d3a-779f-46df-bc7b-1a8c14b78f30	83	
ComScore:6036030	68	VerticalScope
ComScore:6035349	56	NeuLion

All of these pairs of websites are websites with a relation based on webmasters as explained as follows.

The first identifier in Table 3, *ComScore:6036461*, occurs on 27 websites in total. 26 of these websites share a domain name and have a different domain extension, such as tripadvisor.com and tripadvisor.nl. All 26 websites contain the Google Analytics identifier in the second row of Table 3, namely *ua:UA-30198665-1*. The difference between the two clusters is that the biggest cluster contains the extra website oyster.com. It has been verified that this website is owned by TripAdvisor as well[1].

The second cluster is made by two identifiers occurring on the same 7 websites, all owned by StackExchange[2].

The third cluster of websites is a network of publishing websites. The first identifier relates a total of 90 distinct websites to each other, all of them owned by Gatehouse Media[3], except for 6 websites, which are owned by Local Media Group[4]. All of these 6 websites are not included in the cluster of the last identifier (i. e. the *ShareThis* identifier).

1 TripAdvisor: http://www.tripadvisor.com/PressCenter-c6-About_Us.html (Accessed: August 18, 2015)
2 StackExchange: http://stackexchange.com/ (Accessed: August 18, 2015)
3 Gatehouse Media: http://www.gatehousemedia.com/ (Accessed: August 18, 2015)
4 Local Media Group: http://www.localmediagroupinc.com/ (Accessed: August 18, 2015)

This could be an indication that some identifiers are connected to a specific company and some identifiers are used for the overview of all companies. Further investigation in to the relation between Gatehouse Media and Local Media Group is done in which it is discovered that Local Media Group is operated by Gatehouse Media[5], which explains the shared identifier.

In the second to last cluster of websites in Table 3 it is manually verified that all websites are owned by a company called VerticalScope[6]. A majority of websites in the cluster mentions that it is owned by VerticalScope in its footer, in the terms of use or in its privacy statement. The other websites are referred to from VerticalScope's website.

The last cluster of websites is owned by NeuLion[7]. Each website in this cluster contains "empowered by NeuLion" in its footer.

In conclusion it can be said that all websites paired with each other by the ten identifiers from Table 3 represent a webmaster relation, specifically an ownership relation. The majority of these websites could easily be verified by finding a reference to the owner, while for some websites more investigation had to be done. Since about 30% of website pairs are verified it cannot be said, with a hundred percent certainty, that all pairs represent a webmaster relation, but from this 30% percent a 100% is verified as a webmaster relation.

5.2 MACHINE LEARNING APPLICATION

As explained in Section 2.4 it is not always possible to construct an algorithm which is capable of solving a certain problem. Such a problem may be suitable for machine learning as machine learning algorithms are capable of learning from data and solve problems by detecting trends.

In this research machine learning is applied in order to identify relations between websites. A machine learning algorithm is given a dataset of website pairs and their features in order to recognize patterns. These patterns are then used to make a prediction of the existence of a relation between a pair of websites.

To practice machine learning several steps must be undertaken as outlined in Section 2.4. The machine learning process, as applied in this research, is portrayed in this Section.

5 Article about Gatehouse Media operating Local Media Group: http://www.bizjournals.com/newyork/news/2013/09/04/news-corp-sells-dow-jones-local-media.html (Accessed: August 18, 2015)

6 VerticalScope: http://www.verticalscope.com/ (Accessed: August 18, 2015)

7 NeuLion: http://www.neulion.com/ (Accessed: August 18, 2015)

5.2.1 *Ground Truth*

Machine learning is used to find a model which can predict, given two websites, whether they are related. The question that needs to be answered is as follows:

RQ3.A What is the chance that two websites, given their combination of fuzzy evidence, can be correctly identified as a webmaster relation?

RQ3.a is a subquestion of RQ3 which is about identifying relations between websites by considering evidence that can be detected on them.

To be able to provide an answer to RQ3.a, the machine learning algorithm needs to be trained on the data. This training involves informing the algorithm with a label for each pair of websites. This label either indicates that two websites are related or that they are unrelated. The labeled set of website pairs is named the target dataset and forms the desired output of the machine learning algorithm.

This target set is also called the ground truth. Since this set must be provided to the algorithm it must be created before applying machine learning. This process can be tedious and must be done by hand as there is no automated process to identify relations between websites. This labeling can be avoided however by choosing labels other than related and unrelated.

The total number of pairs that can exist is given in Table 6

Instead of the unrelated and related labels, the labels considered are: shared identifier, no shared identifier. Using these new labels the labeling process can be automated as hard evidence can be extracted from websites. More precisely, the new labels must be: shared identifier detected, no shared identifier detected. The reason for this specification is because it cannot be said that this work is capable of detecting all existing identifiers.

The benefit of redefining the labels is that it allows the labeling process to be automated, but it also brings a drawback. This drawback is that the results of the algorithm can not directly provide an answer to RQ3.a. Instead, the following question can be answered:

RQ3.B What is the chance that two websites, given their combination of fuzzy evidence, contain at least one detected shared identifier?

For the rest of this thesis, when speaking about related pairs, it is meant that at least one shared identifier is detected. After gathering and verifying the results, the distinction is made again and the impact of the answer of RQ3.b on RQ3.a is investigated in Section 6.3 using the study in Section 5.1.1.

5.2.2 *Dataset Preparation*

A machine learning algorithm expects, as input, a dataset of feature vectors. Since pairs of websites are considered, the dataset consists of feature vectors of websites pairs. This dataset together with the target set, which as explained in Section 5.2.1 informs the algorithm of the label for each pair, is provided as input for the machine learning algorithm.

For each website a feature vector is generated which is represented by an array of 1's and 0's serving as booleans. The indexes represent the fuzzy evidence features and a value for a certain index is 1 if a feature is detected on a website and 0 if this is not the case. Given enough (discriminating) features it can be said that the feature vector is a website's fingerprint (i. e. it characterizes a website, beyond its URL or IP address).

To be able to create feature vectors for pairs of websites, the feature vectors of two websites must be combined. For two websites, A and B, one feature can appear in 4 forms as shown in the truth table in Table 4.

Table 4: Truth table for one feature on websites A and B

A	B
1	1
1	0
0	1
0	0

Multiple methods can be used in order to combine the feature vectors. Each method has its own benefits and drawbacks. Before applying these methods and evaluating the machine learning results, it is unknown how these benefits and drawbacks affect the results, because the trends in the data are not known.

The following Paragraphs depict the different methods of pairing up feature vectors. For each method its benefits and drawbacks are outlined next to describing the dimensions of the resulting dataset. Truth tables for each merge method are given in Appendix C.

SUM OF SHARED FEATURES The first method that is considered simply counts the number of features shared between both websites. The result of applying this method to two feature vectors is a single number. If this number is high it can be said that websites are similar as they share a great deal of features. This method however does not consider the separate features as it is not capable of encoding all permutations as they occur in Table 4, which means that the resulting feature vector does not cover all information. The motivation behind

this method is that it might be the case that the chance of predicting a related pair increases with the number of shared features. If N is the number of pairs, the dimension of the set of features when using this method is: $N * 1$.

CONCATENATION Another method that can be considered is a method that concatenates both feature vectors. The resulting feature vector is of the form:

$$[F_{a0}, F_{a1}, ..., F_{a(n-1)}, F_{b0}, F_{b1}, ..., F_{b(n-1)}] \tag{8}$$

Where one feature is of the form F_{xi}, where x is the name of the website and i is the index of a feature, with n as the total number of features. From Equation 8 it can be seen that the dimension of the resulting dataset when using this method is $N * 2 * M$, where N is the number of pairs and M is the number of features. The benefit of this method is that all the permutations from Table 4 can be encoded. The drawback however is that information about a shared feature between two websites is not encoded.

CONJUNCTION The conjunction of two feature vectors, results in a dataset with ones for features that are detected in both websites and zeros otherwise. The incentive for this method is that the shared features are more important than the other combinations of features. The benefit of this method is that it is capable of encoding features that are shared between websites. A drawback is that the information about a feature detected on only website is regarded the same as a feature that is not detected on both websites and as such not all permutations can be encoded. This method results in a dataset with a dimension of $N * M$, where N and M are respectively the number of pairs and the number of features. The resulting feature vector is of the following form:

$$[F_{a0} \wedge F_{b0}, F_{a1} \wedge F_{b1}, ..., F_{a(n-1)} \wedge F_{b(n-1)}] \tag{9}$$

ADDITION Another method, with a dimension of $N * M$, where N and M are respectively the number of pairs and the number of features, is the method which sums up both vectors. The possible values that can appear in the resulting feature vector are 0, 1 and 2. 0 represents a feature not detected on both websites, 1 represents a feature detected on one website and 2 represents a feature detected on both. A value of 2 indicates the presence of a feature on both websites and thus this method is capable of encoding shared features. The drawback is that it is not capable of distinguishing between all permutations, because a feature only detected on website A is encoded the same a feature only detected on website B. Another drawback of this method is that features are assigned numeric values[8], which may in-

8 The values 0 and 1 can be seen as booleans

fluence the results because of the assumptions made by the machine learning algorithm. The resulting feature vector is of the following form:

$$[F_{a0} + F_{b0}, F_{a1} + F_{b1}, ..., F_{a(n-1)} + F_{b(n-1)}] \tag{10}$$

CONJUNCTION AND DISJUNCTION Another method that is investigated, is a method which is more complex than the previous methods. This method results in a feature vector with the same dimension as the append method ($N * 2 * M$), but is built differently. The feature vector is, similar to *concatenation*, divided in two parts. One part consists of the conjunction of the two feature vectors, while the other part consists of their disjunction. This results in a feature vector composed as follows:

$$\begin{aligned}&[F_{a0} \wedge F_{b0}, F_{a1} \wedge F_{b1}, ..., F_{a(n-1)} \wedge F_{b(n-1)},\\ &F_{a0} \vee F_{b0}, F_{a1} \vee F_{b1}, ..., F_{a(n-1)} \vee F_{b(n-1)}]\end{aligned} \tag{11}$$

Unlike *concatenation*, this method is capable of encoding shared features. This method does have a drawback however and this drawback is that not all permutations are being encoded. Similarly to *addition*, no distinction can be made for a feature detected on only one of the two websites.

FULL FACTORIAL The last method considered, is a method which combines *concatenation* and *conjunction*. This method does not have any of the drawbacks of the previous methods and it is capable of encoding shared features as well as all permutations. The dimension of the resulting feature vector is $N * 3 * M$, where N is the number of pairs and M is the number of features. This method results in a feature vector of the form:

$$\begin{aligned}&[F_{a0}, F_{a1}, ..., F_{a(n-1)}, F_{b0}, F_{b1}, ..., F_{b(n-1)},\\ &F_{a0} \wedge F_{b0}, F_{a1} \wedge F_{b1}, ..., F_{a(n-1)} \wedge F_{b(n-1)}]\end{aligned} \tag{12}$$

The summary in Table 5 shows the differences in characteristics between the merging methods when looking at the resulting feature vector. The methods are evaluated against two aspects. The first aspect, "All Permutations", indicates whether a distinction can be made between the four different combinations of a feature occurrence as shown in Table 4. With "Encode Shared Features" it is meant that the merge method can encode a feature that is shared between two websites.

Although some methods perform better than others, according to the table, this might not be an indication of their actual performance. The impact of the benefits and drawbacks of the different methods depends on the trends in the data. That is, it might be the case that the information that is lost through merging does not contribute to the detection of trends in the data.

Table 5: Overview of characteristics of merge methods considered in dataset preparation

MERGE METHOD	ALL PERMUTATIONS	ENCODE SHARED FEATURES
Sum of shared features		×
Concatenation	×	
Conjunction		×
Addition		×
Conjunction and disjunction		×
Full Factorial	×	×

Another detail to notice is that Table 5 is a high level comparison of the merge methods. Therefore, it is important to know that the presence (or absence) of crosses can be interpreted differently per merge method. An example of this is the difference in degrees of being able to encode all permutations. More specifically, the first merge method (*sum of shared features*) is not capable of encoding any permutation while the *addition* method is able to at least encode some permutations. Another example is that *sum of shared features* is able to encode shared features, but unlike other methods, being able to encode shared features, this method does not give insight in to which features are shared.

5.2.3 *Algorithm*

Machine learning can be applied by various algorithms. This research applies logistic regression instead of other algorithms. In this Section a description about relevant machine learning algorithms is given along with a motivation of choosing logistic regression over the other algorithms.

Section 2.4.2 describes that a machine learning algorithm can be separated on its learning method or its output.

Dividing the algorithms on their learning method results in supervised, unsupervised and reinforcement learning algorithms. The algorithms suitable for this research are supervised algorithms, because input and output should be considered when creating a model. The other two categories cannot be used since unsupervised algorithms only consider input and reinforcement learning algorithms produce sequence of actions as output.

Dividing the algorithms on their output results in classification, regression and clustering algorithms. The goal is to be able to predict whether two websites are related and therefore the algorithms that are used are classification algorithms. The other two algorithm types

are unsuitable, because regression algorithms predict a number from input and clustering algorithms label input in groups which are not known in advance.

5.2.3.1 *Description of Algorithms*

The following Paragraphs describe four supervised classification algorithms, in which the definitions are derived from *Introduction to Machine Learning* by Alpaydin (Alpaydin, 2010).

k-NEAREST NEIGHBORS k-Nearest Neighbors (k-NN) is an algorithm which is capable of assigning the input a class which corresponds with the class given to the most similar feature vectors. These similar feature vectors are called neighbors and the number k in k-NN represents the number of neighbors to be considered. With a value 1 for k the class that is assigned to the input is the class of the nearest neighbor.

LOGISTIC REGRESSION Logistic regression models the relationship between independent and dependent variables. This algorithm can appear in three forms: binomial, multinomial or ordinal. Binomial logistic regression considers output that is divided in two classes. In multinomial logistic regression the output consists of three or more classes and in ordinal logistic regression the output is ordered. Logistic regression creates a mathematic equation by using the independent variables. This equation is of the form:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n \tag{13}$$

where

β_0 intercept

β regression slope coefficient

X value of a feature

The logistic function is applied on Y in order to map the resulting probabilities between zero and one. The resulting equation is then as follows:

$$Y = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n)}} \tag{14}$$

The output Y is the probability that the dependent variable is a value of 1, also called success.

NAIVE BAYES CLASSIFIER A naive Bayes classifier uses labeled examples and probabilities for classifying its input. This is done by using Bayes' rule which, in words, can be explained as the process of predicting the class of the input by using a known outcome for a certain feature. Bayes' rule is only applicable for one feature and

therefore, to be applicable to multiple features, the naive Bayes classifier considers its features to be independent. To summarize, a naive Bayes classifier considers features to be able to assert how likely the input is of a certain class by looking at previous results.

SUPPORT VECTOR MACHINE Support Vector Machine (SVM) can be used to create a model in which the input is represented as points in a space. A decision boundary is drawn which separates these points in distinct classes. The decision boundary maximizes the gap between the classes. New input is inserted as new points and for each point it is checked on which side of the gap it is located and thus it can be established to which class the point belongs to. Points that define the boundaries of the gap are called the support vectors and are the only points that encode information. Other points can be left out without affecting the results.

For this work the logistic regression algorithm is used. The most important reason for using this algorithm is that it is simple and gives insight in to the effect of separate features. A naive Bayes classifier is also capable of giving insight in to the effect of individual features, but considers these features to be independent, which might not be the case. The k-NN algorithm, although simple, is not capable of giving insight in to separate features as well as the more complicated SVM.

5.2.3.2 *Application of Logistic Regression*

The previous Section describes various machine learning algorithms and gives a motivation for using logistic regression. This Section describes, in general, the pipeline of applying logistic regression on a dataset. This pipeline is executed a number of times in order to calculate average results.

Firstly, a dataset is created from all related pairs of websites that can be identified in the dataset. Two websites are related, in this context, when they share at least one identifier.

Subsequently, a dataset is created exclusively containing unrelated pairs. A pair is considered to be unrelated if no shared identifier is detected. This set of unrelated pairs is equal in size to the set of related pair as to prevent the machine learning algorithm from being biased.

Where all related pairs are considered with each execution of the pipeline, a subset of all unrelated pairs is considered, because in general the number of unrelated pairs is substantially greater than the number of related pairs. In each execution of the pipeline a random subset of all unrelated pairs is considered in order to apply the algorithm on several parts of the dataset.

All unrelated and related pairs, considered in the pipeline, are then placed in one dataset. Using this dataset, six datasets with features are

generated each created by one of the six merging methods described in 5.2.2. These datasets of features are also called X or example inputs.

Along with these six datasets another dataset is created which is called y, the target dataset or the desired output. This dataset consists of 0 and 1 values which represent the classes of pairs. A class with a value of 1 is considered to be the class that needs to be predicted. In this case, a related pair is labeled with a 1 and an unrelated pair is labeled with a 0. This dataset serves as the ground truth as described in Section 5.2.1.

Thus for a pair of websites, its features are encoded at an index in the dataset of features X and its label is defined at the same index in the target dataset y. These datasets are each split, at the same point, in a train and a test set.

The train set is used for fitting a model and thus serves as the set from which trends are extracted. The test set serves as a set on which the trained model is applied and can be used to measure the model's performance.

Thus four datasets are used when applying a machine learning algorithm in this case. X and y training sets and X and y test sets.

The pipeline is concluded by measuring the model's performance using the test sets and F_1 scores. F_1 score is a performance measure considering precision and recall and is described in more detail in Section 2.4.3.

5.3 TOOLS

This Section describes tools, worth mentioning, that are used as well as why a tool is needed and how it is used.

5.3.1 *Apache Hadoop*

Apache Hadoop is an open-source MapReduce framework that is capable of processing large amounts of data (White, 2009; Hadoop, 2015). Hadoop is able to partition data which allows for distributive and parallel computing. Reliability is ensured by identifying and dealing with failures at the application layer.

The partitioning of data across multiple machines is made possible by the Hadoop Distributed Filesystem (HDFS) (White, 2009; Hadoop, 2015; Shvachko et al., 2010). The HDFS is integrated in Hadoop and operates two types of nodes, the master and the worker. The master node, named *namenode*, is responsible for the filesystem hierarchy and metadata for all files and directories. Additionally, the namenode is aware of the location of file blocks scattered across the datanodes. The worker nodes, named *datanodes*, store and retrieve file blocks and update the namenode with the whereabouts of the blocks.

The storage of big data is facilitated by HBase, a column oriented NoSQL database build on top of the HDFS (White, 2009; Hadoop, 2015; George, 2011). HBase scales linearly by adding nodes as it is built with scalability in mind.

The CAP-theorem states that it is impossible for a distributed system to simultaneously guarantee more than two of the following three guarantees (Brewer, 2000):

- consistency;
- availability; and
- partition tolerance.

HBase is able to provide the guarantees of consistency and partition tolerance (George, 2011). More specifically, HBase is strongly consistent, which means that changes in data are atomic and seem to occur almost immediately.

Another key component of Hadoop is the MapReduce paradigm. This paradigm consists of two phases, the map phase and the reduce phase (White, 2009; Dean and Ghemawat, 2008). The input and output of the map and reduce phases are key-value pairs. Consider the example of counting words in a text. In this example, the map phase maps a value of one to each word encountered. Subsequently, the reduce phase aggregates the output of the map phase, resulting in a list containing unique words and for each word its number of occurrences in the text.

The source code of the word count example can be found on: `http://hadoop.apache.org/docs/current/hadoop-mapreduce-client/hadoop-mapreduce-client-core/MapReduceTutorial.html` (Accessed: September 1, 2015)

Hadoop is used in order to be able to scale the extraction of identifiers and features from websites to multiple machines. The raw websites are stored in HBase along with the metadata and intelligence extracted from the websites. Examples of information, besides the identifiers and features, that can be extracted are languages, countries and email addresses.

5.3.2 *Elasticsearch*

Apache Lucene is a library, written in Java, which provides full-text search functionalities (Lucene, 2015). This means that Apache Lucene, when integrated in a Java program, provides search and index functionalities for large datasets. However, this integration is complex and for this purpose Elasticsearch is written (Gormley and Tong, 2014). Elasticsearch is an open source search engine built on top of Apache Lucene and is capable of hiding Lucene's complexity behind an easy to use API.

Elasticsearch is used to be able to search the information that is extracted from websites.

5.3.3 *Graph Visualization*

To be able to visualize a network of websites a graph is used. This graph is build using several tools, each responsible for different roles as outlined in this Section.

Storage of the data that form the graph is managed by Titan (Thinkaurelius, 2015). Titan is a distributed graph database which is capable of storing and querying a large number of vertices and edges. Titan can easily be integrated with Hadoop, Elasticsearch and HBase.

Titan also provides a great integration with Gremlin, which serves as means of communication with the graph (Rodriguez et al., 2015). Gremlin provides a language which allows for querying, manipulating and traversing the graph data.

Next to the storage of data and interaction with it, the graph needs to be visualized. This visualization is made possible by the KeyLines technology (Intelligence, 2015). KeyLines provides a technique which is capable of visualizing a graph from data stored in a Titan graph database and allows the user to manipulate this graph. KeyLines offers a wide range of features, such as visualizing data on maps and filtering data on time and date. These and other features of KeyLines are aimed at increasing a user's insight in the data.

The result of all these tools is shown in the graph of Figure 3.

5.3.4 *VoyagerOne*

Web-IQ, the company supervising this work, develops a tool called VoyagerOne. This tool encompasses, crawling the (public) Internet, analyzing data and visualizing analyzed data.

This product is used in order to get insight in data and among others solve law enforcement cases.

5.3.5 *Scikit-learn*

Scikit-learn is a Python library which provides machine learning algorithms and numerous helper functions (Pedregosa et al., 2011). The library is well documented and easy to use. The main advantage of using this library is that the machine learning algorithms and other functionalities do not have to be custom made. This results in being able to put emphasis on utilizing the algorithms. More importantly, the algorithms are less likely to contain errors.

5.4 SUMMARY

Identifiers and features are extracted using pattern matching and string comparisons. Appendix A and Appendix B respectively list the extracted identifiers and features.

A study is performed in order to investigate the meaning of a shared identifier for identifying a webmaster relation between websites. The results of the study suggest that a shared identifier is an indication for a webmaster relation (i. e. ownership relation).

The ground truth that is established for machine learning is based on RQ3.b. Six methods of preparing data are created and are compared. From the four machine learning algorithms described, logistic regression is applied on the datasets created by these six methods.

Finally, the tools that are used are described. From these tools Apache Hadoop is used to be able to apply the MapReduce framework across multiple machines. Elasticsearch provides searching functionality for the information extracted from websites. Visualization of a network of websites is made possible by the graph database called Titan, the query language Gremlin and the visualization framework KeyLines. All these tools are brought together in the VoyagerOne interface. Lastly, Scikit-learn is described which provides various libraries for machine learning.

6 EVALUATION AND RESULTS

In order to gain insight in the results, the dataset used is elaborately described in Section 6.1.

Following this description is Section 6.2 in which the results are presented and analyzed.

The work is discussed in Section 6.3, followed by an overview of tasks that are left for the future in Section 6.4.

6.1 DATASET

A dataset of websites is needed to be able to verify the ideas of this research. Such data should not be biased to a specific part of the Internet as for it to be representative. One could create such a dataset oneself and spent time and thought in to selecting seeds, avoid spam or websites using Search Engine Optimization (SEO).

As explained in Section 2.2, a crawl is initiated by selecting websites, called seeds, which serve as the starting point of the crawl. Knowing what the dataset will be used for is essential when choosing seeds. If a country specific dataset is to be created, one could start crawling from the list of most popular websites for a country as provided by Alexa[1]. However, starting to crawl from this list does not guarantee that all websites for a certain country will be found. Similarly, one can use Alexa's list of 500 most popular sites on the web for a certain dataset, but there is no guarantee that this dataset is representative to the Internet.

Other obstacles that one can face during crawling are spam and SEO. Spam and SEO are methods used for creating profit in commercial websites and are usually not of interest to Internet users (Fetterly et al., 2004). To avoid having these websites pollute the dataset of crawled websites, a great deal of effort should be invested in the web crawler.

Instead of creating a dataset of web crawl data, an existing dataset is used in order to avoid the aforementioned hurdles. The dataset used for testing and evaluation purposes is obtained from the Common Crawl (Foundation, 2015). The Common Crawl, from the Common Crawl Foundation, is a non-profit initiative of volunteers to create a publicly available dataset containing crawled websites. Common Crawl Foundation's aim, by making the Common Crawl publicly available, is that everyone can focus on research and analysis

1 Alexa top 500 most popular sites: http://www.alexa.com/topsites (Accessed: September 1, 2015)

instead of gathering data. Furthermore, they aim at having a dataset which is of high quality and avoid "... webspam, porn, and the influence of excessive SEO (search engine optimization)"[2].

According to their website, the Common Crawl consists of data gathered over the past seven years. The data includes billions of webpages and trillions of links and the size is several petabytes. New data is made publicly available each month and crawled data is partitioned in days.

For the purpose of this research a part of one day of crawling is considered, the 17th of April 2015[3]. Extracted, this part amounts to 5 gigabytes of storage.

6.1.1 *General Statistics of the Dataset*

Basic information about the dataset is given in Table 6.

Table 6: Table of general statistics on a piece of data in the Common Crawl

TYPE OF INFORMATION	VALUE
Number of websites	20.416
Number of possible pairs[4]	208.396.320
Number of related pairs	28.033
Number of pages (URL)	59.215
Number of countries (mentioned in text)	443.030
Number of languages (mentioned in text)	58.212
Number of unique domain extensions	131

The top ten largest websites is given in Table 7. A network revolving around these websites is shown in Figure 3a.

Figure 3a shows a graph of the ten largest websites in the dataset as visualized in VoyagerOne (see Section 5.3.4). The websites are visualized with orange circles, called vertices. The orange vertices with a blue highlighting indicate the largest websites while the other orange vertices are websites which are highly connected to the ten largest websites. Connection, visualized with arrows called edges, represent hyperlinks between the websites. The blue vertices are identifiers from hard evidence and an edge between a website and an identi-

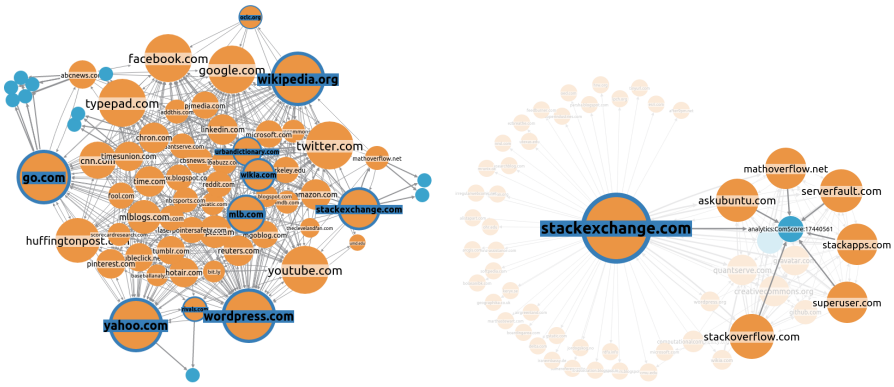
2 http://blog.commoncrawl.org/2012/12/blekko-donates-search-data-to-common-crawl/ (Accessed: September 1, 2015)

3 The data can be downloaded through the following link: https://aws-publicdatasets.s3.amazonaws.com/common-crawl/crawl-data/CC-MAIN-2015-18/segments/1429246633512.41/warc/CC-MAIN-20150417045713-00039-ip-10-235-10-82.ec2.internal.warc.gz (Accessed: September 1, 2015)

4 All possible combinations can be calculated by the formula given in Equation 6

Table 7: Overview of largest websites with their number of pages

WEBSITE	NUMBER OF PAGES
wikipedia.org	500
urbandictionary.org	347
stackexchange.com	335
wordpress.com	259
mlb.com	189
wikia.com	120
yahoo.com	116
go.com	114
oclc.org	101
rivals.com	97



(a) Graph of largest websites in the Common Crawl dataset

(b) Graph of websites centered around stackexchange.com

Figure 3: Visualization of a part of the network of websites in the Common Crawl dataset

fier indicates that the website contains the identifier[5]. In the Figure on the right it can be seen that stackexchange.com and mathoverflow.net have hyperlinks to each other. Besides this relation, they are related by two separate identifiers as indicated by their edges to the blue vertices. Further investigation learns us that both websites are owned by the company Stack Exchange which owns over 130 Q&A communities[6].

More insight in to the network around stackexchange.com can be given by generating a graph centered around this website. Figure 3b visualizes the network, as stored in the dataset, that involves stackex-

5 Identifiers are only shown in the graph when they are detected on at least two websites

6 Stack Exchange: `http://stackexchange.com/about` (Accessed: September 1, 2015)

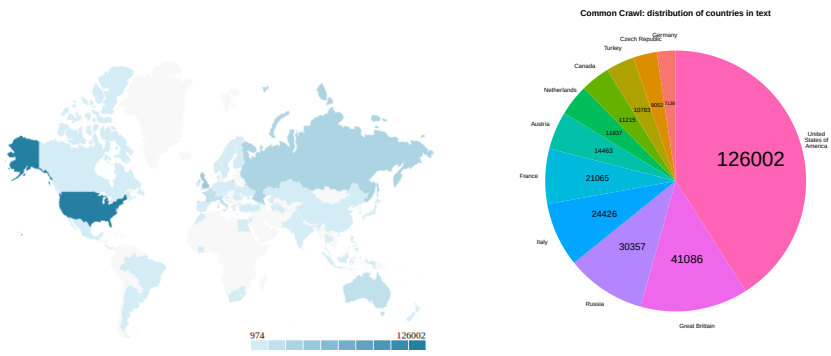
change.com. It can be seen that stackexchange.com refers to a number of small websites and that there are a total of 7 websites that share identifiers. All of these 7 websites are part of the network of websites owned by Stack Exchange.

The top ten of most encountered languages in text is displayed in Table 8. English is, by far, the most encountered language.

Table 8: Overview of languages encountered the most in text

LANGUAGE	NUMBER OF DETECTIONS
English	53610
Spanish	898
German	585
French	554
Portuguese	329
Italian	242
Indonesian	172
Polish	168
Turkish	140
Dutch	123

In addition to this top ten, information about encountered countries in text is given in Figure 4. Figure 4a shows a map of countries where each country is colored depending on the number of its occurrences in text. Furthermore, Figure 4b visualizes the occurrences of countries in text which are mentioned the most.



(a) Geographical distribution of countries mentioned in text

(b) Pie chart of the distribution of countries mentioned in text

Figure 4: Visualization of the distribution of countries mentioned in text

In Figure 5 an overview of the occurrences of the domain extensions encountered the most is given. From this pie chart it is clear that most domain extensions are ".com" in this dataset.

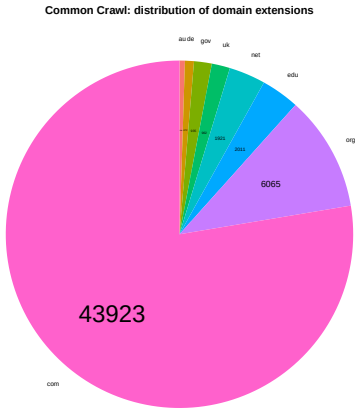
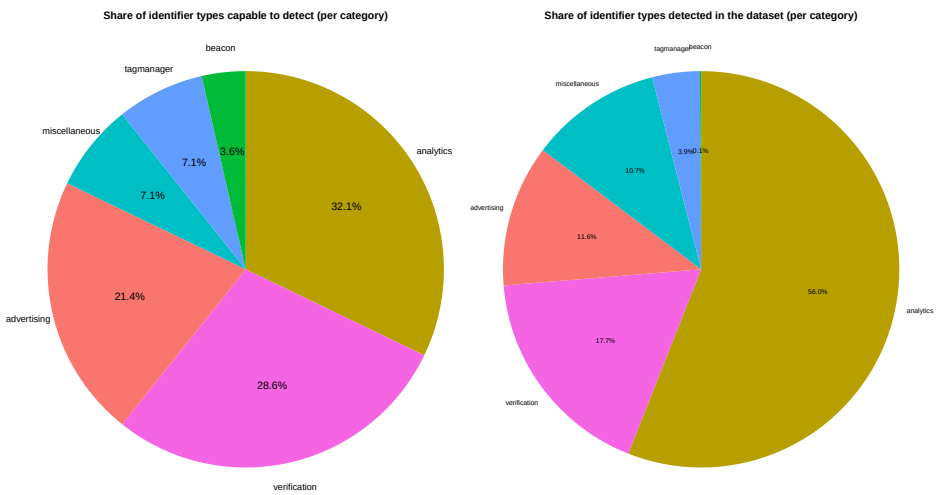


Figure 5: Pie chart of occurrences of domain extensions

6.1.2 *Evidence in the Dataset*

Figure 6a, which is a duplicate of Figure 2 in Section 5.1, shows the share of the implemented identifiers over categories. The pie chart in Figure 6b visualizes the share of each identifier type per category in the dataset. From a comparison of these charts it can be said that the share of a category reflects the share of identifiers implemented per category. For the beacon category, only one identifier is implemented and this identifier does not occur often in the dataset. The most identifiers implemented are from the analytics category which also occur the most in the dataset.



(a) Share of implemented identifier types per category

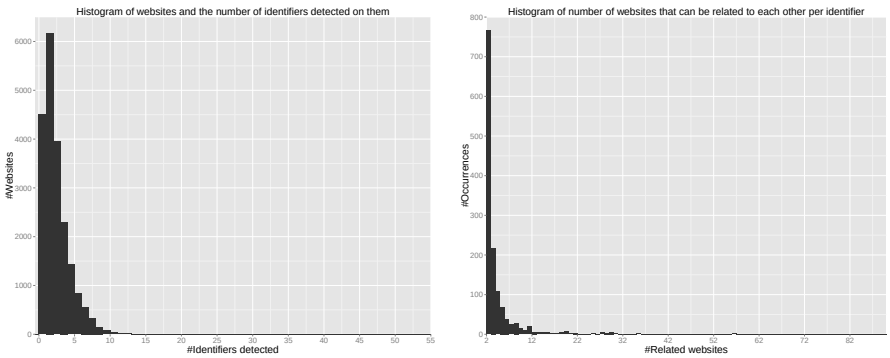
(b) Share of identifiers in the dataset per category

Figure 6: Share of identifiers

Figure 7a shows a histogram of the number of identifiers that are detected on a website. It is important to notice that from the 20.416

websites, a majority of the websites contain identifiers. Around 6.000 websites have 1 identifier detected and approximately 4.000 websites have 2 identifiers detected on them. No identifiers are detected on 4.500 websites, which is around one fifth in total. It is calculated that the median and mean are both 1 identifier with a total of 39.977 identifiers detected on 20.416 websites.

Figure 7b is a histogram of the number of websites that can be related to each other per identifier. It can be seen that the vast majority of identifiers are detected on two websites.



(a) Number of identifiers detected per website

(b) Occurrences of the number of websites than can be related with each other per identifier

Figure 7: Identifier histograms

From Figure 8 it can be seen that *analytics:ua*, which is Google Analytics, is the identifier which is detected the most followed by *miscellaneous:FacebookSDK* and *verification:GoogleSiteVerification*. From the Figure it can be seen that each identifier is detected at least once.

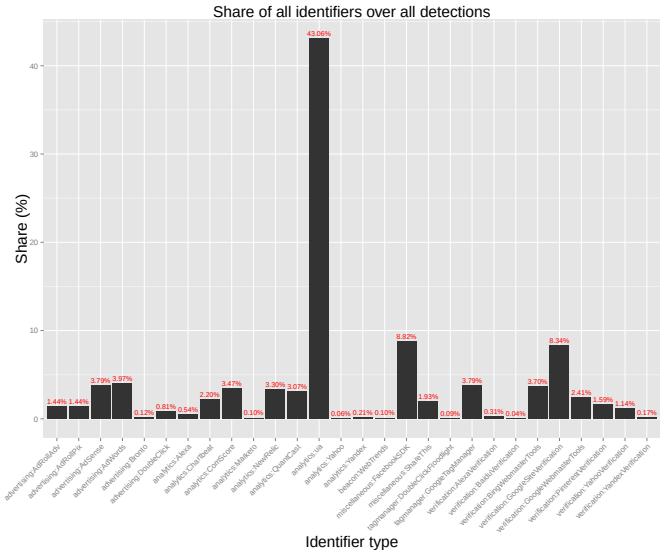
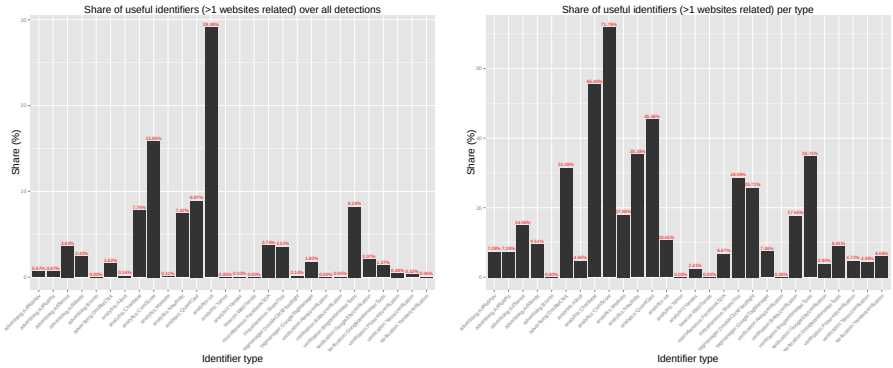


Figure 8: Share of identifier types over all identifier detected

While the chart in Figure 8 is important for an overview of the share of each identifier type, it cannot give information about what each identifier contributes to finding relations between websites. It might for example be the case that none of the *analytics:ua* identifiers is detected on multiple websites.

To gain insight in to the identifiers that have been detected on multiple websites, the graphs in Figure 9 are used. These graphs consider the share of identifier types over all identifiers that are detected on multiple websites. From Figure 9a it can be seen that the distribution is similar to Figure 8. Some identifiers, such as *analytics:Chartbeat* and *analytics:ComScore* are not detected often on websites, but from these detections a large share is encountered on multiple websites. It can also be seen that there are identifiers which were never found on multiple websites. All these identifiers, which did not relate websites to each other, have a small share in Figure 8.



(a) Share of identifiers detected on multiple websites over all identifiers

(b) Share of identifiers detected on multiple websites per identifier type

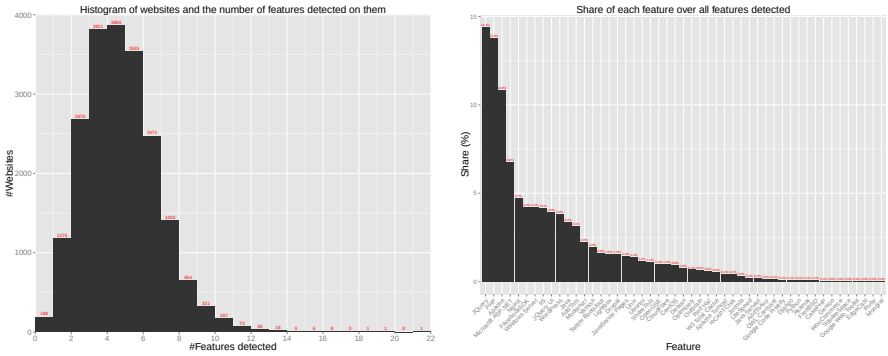
Figure 9: Share of identifiers detected on multiple websites

Figure 9b can be used for reasoning about the usefulness of an identifier per type. With usefulness, identifiers relating multiple websites to each other are meant. This Figure supports the previous statement about identifiers types, such as *analytics:Chartbeat* and *analytics:ComScore*, having a large share of useful identifiers. It can also be seen that only 10 percent of all *analytics:ua* identifiers are useful in the dataset.

Just as Figure 7a shows the number of identifiers detected on a website, Figure 10a shows the number of features detected per website. From this Figure it can be concluded that no features are detected on less than 200 websites. Three, four or five features are detected on the majority of websites in the dataset. It is calculated that the median and mean are both 4 features with a total of 87.433 features detected on 20.416 websites.

Figure 10b visualizes the share of each feature in all detected feature sorted on their share. It can be seen that *JQuery*, *PHP* and *Apache*

are detected the most. With one detection the features *Mongrel* and *Ruby* are detected the least.



(a) Number of features detected per website

(b) Share of a feature over all features detected

Figure 10: Statistics about features detected in the dataset

6.2 RESULTS

A logistic regression algorithm is applied on a dataset as described in Section 6.1. The results are measured by F_1 scores of the six merging methods described in Section 5.2.2 (more information about F_1 scores is given in Section 2.4.3). The F_1 scores are compared and a more detailed investigation is performed on the merging method performing the best.

The logistic regression algorithm is trained on a subset of the dataset described in Section 6.1. The reason for this is the expectation of the input for the logistic regression algorithm. The process of applying a logistic regression algorithm on a dataset is explained in Section 5.2.3.2.

The results are gathered by executing the machine learning pipeline from Section 5.2.3.2 a hundred times. Each execution considers the same 28.033 related pairs and 28.033 randomly chosen unrelated pairs. The unrelated pairs are chosen randomly in order to address several parts of the datasets with several executions. The total number of unrelated pairs that can be created in the dataset is around 200 million.

In order to apply cross validation, the datasets are split in to train and test sets, in which the train sets amount to 80% of the data and the test sets to 20%.

6.2.1 F_1 scores

The resulting F_1 scores, averaged over 100 executions, are shown in Figure 11 with standard deviation.

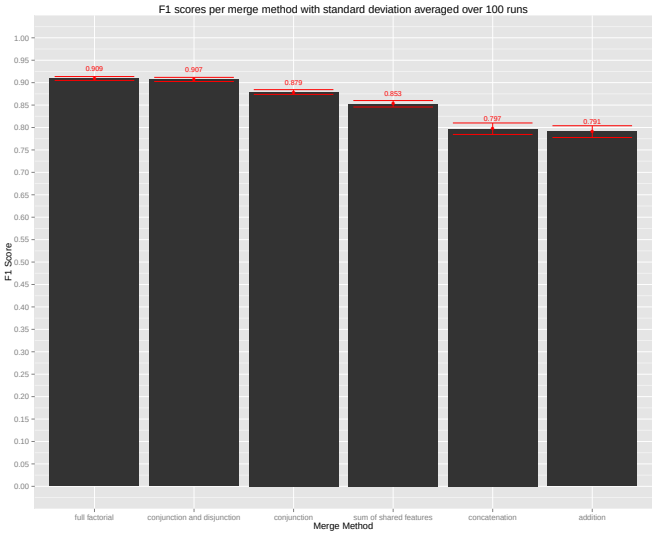


Figure 11: F_1 scores of merging methods

With F_1 scores around 0.79, the merging methods *concatenation* and *addition* perform the worst of all methods. It cannot be said with a hundred percent certainty which method performs better as the results are close to each other and the standard deviations overlap. An F_1 score of 0.79 can be interpreted as being able to predict, given two websites, that they contain a shared identifier with a 79% correctness.

The *sum of shared features* and *conjunction* methods perform better and the F_1 score of *conjunction* is higher by a small margin.

Full factorial and *conjunction and disjunction* are the merging methods performing the best with an F_1 score of almost 0.91.

The results of the separate merge methods can be explained by looking at their benefits and drawbacks which are outlined in Section 5.2.2.

The merging method with the lowest F_1 score is *addition*. This method has the drawback of not being able to encode all permutations, but since other methods have this drawback as well this does not explain the fact that this method performs the worst. The reason for performing the worst is possibly the unique aspect of this method using numeric values instead of booleans for encoding its features.

Concatenation, although being able to encode all permutations, is also one of the worst methods. It is suspected that this is the result of not being able to encode shared features, since *concatenation* is the only method not being able to encode shared features in some way.

The *sum of shared features* method loses all information regarding individual features, as the feature vector consists of one value. This drawback seem to be outweighed however by being able to encode shared features, since this method performs better than the previous two methods. The F_1 score of this method is 0.85, which might indicate that related websites are similar in the sense that they have the same features.

As opposed to *sum of shared features*, *conjunction* can give insight in to separate features, resulting in losing less information when merging. This is also seen in the performance of these methods.

Conjunction and disjunction is a method which can encode even more information than *conjunction*. This trend is also seen in the results. Where *conjunction* encodes a feature not detected on both websites the same as a feature detected on one website, *conjunction and disjunction* encodes this separately.

Lastly, the best method *full factorial* is able to encode all information, which explains why it performs the best of all merging methods.

6.2.2 *Full Factorial*

In Figure 11 it can be seen that the *full factorial* merging method gives the best results of all six merging methods. In this Section the results of *full factorial* are given a closer look.

Section 5.2.3.1 provides an explanation of the logistic regression algorithm. Logistic regression works by finding a formula which produces the best results. This formula is made out of coefficients and values for features. The features are served as the input and the optimal regression coefficients are found.

Since the dimension of a dataset merged with *full factorial* is $N * 3 * M$, where N is the number of pairs and M is the number of features, there are $3 * M$ coefficients. The list of features is divided in three parts as seen in Equation 12. The first part consists of the M features of the first website in the pair. The second part consists of the M features from the other website. Lastly, the third part consists of M features which are conjunctions of the features for both websites. The first two parts encode individual features for both websites, while the last part encodes shared features, hence *full factorial* is a combination of *concatenation* and *conjunction*.

Figure 12 gives an overview of the coefficients from the last part of the coefficients when applying *full factorial*. An overview of all $3 * M$ coefficients is given in Appendix D.

From the coefficients in Figure 12 it can be seen that the feature *Debian* contributes the most to the model when detected on both websites. A few features, such as *Mongrel* and *Gentoo* seem to have a negligible influence. The feature *JavaServer Pages* contributes negatively. Furthermore, there are two pairs of features with the same coefficients and standard deviations. One pair is *Python* and *Django*. The reason for this is that *Python* is only detected when detecting *Django*. The other pair is *Mongrel* and *Ruby* as the latter is only detected when *Mongrel* is encountered.

To be able to provide a more in-depth explanation of the coefficients, the graph in Figure 12 is compared with the share of the features in the dataset. This comparison is done in Figure 13.

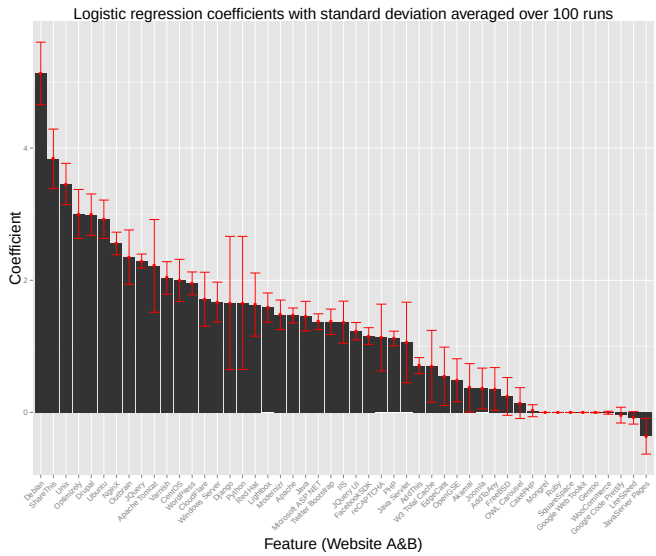
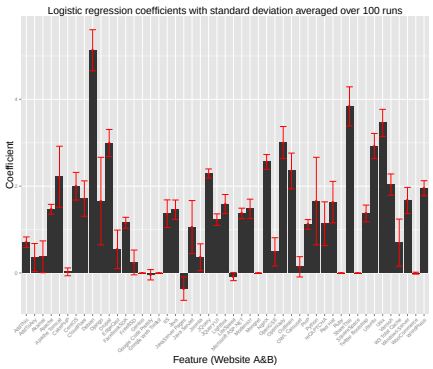
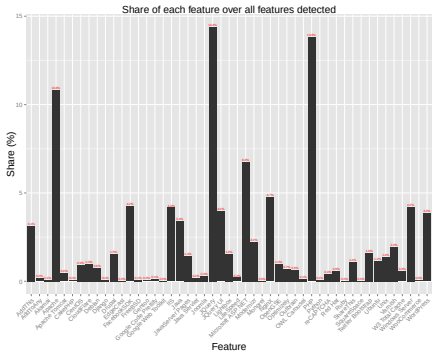


Figure 12: Logistic regression coefficients for *full factorial* (shared features)



(a) Lexicographically sorted logistic regression coefficients for *full factorial* (shared features)



(b) Lexicographically sorted share of a feature over all features detected

Figure 13: Comparison of coefficients with the share of features in the dataset

From the comparison it can be seen that the features with the highest coefficients have a low share in the dataset. For example, the feature *Debian* has a share of 0.8% and is the highest coefficient. The same holds for features such as *ShareThis* and *Unix* with respectively a share of 1.1% and 1.4%. From these features, with a low share in the dataset and a high coefficient, it can be said that they are mostly detected on websites that are related to each other in the dataset.

Other features with a low share, such as *SquareSpace* and *LiteSpeed* have low coefficients from which can be concluded that they are mostly not detected on related websites.

From the five features with the highest share, *JQuery*, *PHP*, *Apache*, *Microsoft ASP.NET* and *NginX*, it is seen that they positively contribute to the model with coefficients between 1 and 2.5 with small standard deviations. This means that although these features are de-

tected on a large number of websites, they still provide valuable information for identifying relations.

6.3 DISCUSSION

The results in Section 6.2 are applied on a relatively small part of data. Although different executions of a machine learning pipeline apply logistic regression on different parts of the dataset it cannot be said that the results are representative for the Internet. From the small standard deviations in Figure 11 it seems that the results do not differ greatly for each piece of data considered, but it is unknown how the results are affected by other parts of the Internet. The volunteers of Common Crawl have put time and effort in creating a dataset of high quality which addresses different parts of the Internet. This means that the dataset is not influenced by spam, SEO and does not include porn. On the Internet however these phenomena do occur and they might influence the results.

Besides the size of the data considered, the results may be influenced by the distribution of types of website pairs. I only train and test the model on an unbiased set of pairs in which the number of related pairs is equal to the number of unrelated pairs. For increased reliability, the model must be tested on biased datasets, specifically datasets containing more unrelated than related pairs. The reason for this is that Internet mainly consists of unrelated pairs. Table 6 shows that there are a total of 28.033 related pairs against around 0.2 billion unrelated pairs. Since not all identifiers are extracted and given the fact that not all related websites have to have shared identifiers, this number of related pairs is a minimum. Despite not being able to calculate the exact number of related and unrelated pairs it is save to say that the number of unrelated pairs greatly outweighs the number of related pairs.

In order to automate the labeling of website pairs for machine learning, it is established that the ground truth is based on a detected shared identifier instead of whether a pair is related by a webmaster. This process is automated as to prevent having to label pairs manually and thus more data can be considered. The impact of this redefinition is that RQ3.a cannot be directly answered. Instead, this question is reformulated in to RQ3.b which is answered in Section 6.2. To be able to provide an answer to RQ3.a, and implicitly to RQ3 as well, an investigation is performed on the meaning of a shared detected identifier for the occurrence of a webmaster relation in Section 5.1.1. In this investigation, 30% of the pairs labeled as related are considered and verified to represent ownership relations and thus webmaster relations. Since all pairs are webmaster relations there is a big chance that detecting a shared identifier is an indication for a webmaster relation and that

the model developed for shared detected identifiers is similar to one for webmaster related pairs.

With an F_1 score of up to 0.91 the developed model is highly accurate. No work is found with a similar approach to which this work can be compared. It is suspected that search engines, such as Google, utilize the information from identifiers and features, but no documentation is found on the subject. Since there is no other work to compare to, future work must be performed on improving the results as described in Section 6.4.

6.4 FUTURE WORK

Future work is divided in to three parts. One part concerns the work that can be done on improving the results. Another part is focused on enhancing the reliability. The last part is dedicated to work on answering questions not answered with this work. Besides these parts, work must be performed on applying the fitted model in the VoyagerOne software.

6.4.1 *Enhancement of results*

Logistic regression is the only machine learning algorithm that is applied. Although the algorithm is powerful, other algorithms might be able to produce a model with a higher accuracy. One of the algorithms to be considered is k-Nearest Neighbors (k-NN). k-NN can be applied by comparing a feature vector for a pair of websites with feature vectors of all other pairs. The label assigned to the pair that is considered is the label assigned to the majority of neighbors. The number of neighbors considered depends on the value of k for which different values must be explored. Other algorithms to be explored are a naive bayes classifier, which uses probabilities for features, and the Support Vector Machine (SVM), which tries to separate website pairs in distinct classes.

Next to the previously mentioned machine learning algorithms an algorithm can be devised which takes advantage of the large number of unrelated pairs on the Internet. For a website pair, not containing a shared identifier, this algorithm always labels a pair as unrelated, since the number of unrelated pairs is much greater than the number of related pairs on the Internet. The precision and recall can be collected for a dataset representative for the Internet and can be combined in to an F_1 score. This F_1 score serves as a baseline for the machine learning algorithms. Every algorithm not performing better than the baseline must be improved or disregarded.

Besides exploring other algorithms, which might improve results, future work can focus on extracting more identifiers and features. The extraction of more identifiers leads to the identification of more

related websites which will form a more reliable labeling for machine learning. This also means that extraction of identifiers brings RQ3.a and RQ3.b closer to each other. If more features are extracted from websites it is expected that the results are enhanced as more information about websites is available. Besides the extraction of more features, different types of features can be extracted. All of the 49 features extracted in this work are technologies, such as programming languages and servers. Besides technologies a plethora of other features can be extracted from websites, such as visual features (e. g. colors, structure) and content (e. g. most used words, email addresses).

6.4.2 *Enhancement of reliability*

In order to assess and increase the reliability of the results, overfitting must be prevented by fitting the logistic regression model on other parts of the Common Crawl dataset and other datasets of the Internet.

Besides more coverage of the Internet, the model must be tested against datasets that differ in the share of related pairs. For the model to work on the Internet it is important that it is verified against datasets biased towards unrelated pairs.

6.4.3 *Unanswered Questions*

Section 6.4.1 mentions the extraction of more features in order to improve results. Investigation must be performed in to the impact of detecting more features on the results. Ultimately, a set of features should be selected which produces the best results.

Another interesting investigation is the correlation between features. There are some features, such as Python and Django which belong together in the sense that when Django is detected, Python is detected as well. Some features are mutually exclusive, such as servers. Although two websites might still be related with different servers, there might be features which are never detected together on two websites. An example of this might be that a webmaster will never use the programming language Ruby when it is accustomed to Python.

Future work can also be dedicated to finding differences in sites of which a great deal of features are similar. For example, there is a service called Blogger[7] in which users can write blogs and create websites using existing or custom templates. For all the blogs with one of the default templates, the number of visual similarities is large. To be able to distinguish between the owners of the individual blogs hints have to be extracted from the content, possibly using authorship

7 Blogger: `https://www.blogger.com` (Accessed: September 18, 2015)

identification. Similarly to Blogger, Wordpress[8] offers a service which allows users to create websites optionally using default templates.

6.5 SUMMARY

The Common Crawl dataset provides a large, high quality, corpus of web data to be used, among others, for academical purposes.

A small part of this dataset is used for fitting a model with logistic regression. Six datasets are created during preprocessing which are made with different methods for merging feature vectors. These datasets serve as input for logistic regression of which the results, measured with an F_1 score, vary between 0.79 and 0.91.

These results, in combination with the study in Section 5.1.1, answer RQ3, which is about combining evidence in order to identify relations between websites. The last research question (RQ4), about the evaluation of the results is answered as well. The evaluation is performed with a cross validation of the fitted model.

The answers to RQ3 and RQ4 do only hold for a small part of the Common Crawl dataset. Future work on larger and different datasets must be performed in order for the work to be representative for the Internet. Besides increasing the reliability, work must be performed on enhancing the results. This can be done by applying different algorithms and increasing the number of identifiers and features that can be detected.

8 Wordpress: https://wordpress.com/ (Accessed September 18 2015)

7 CONCLUSION

There are different types of relations between websites, such as:

- websites with identical authors;
- websites with similar functionality or content;
- relations based on hyperlinks;
- relations based on a user's browsing behavior;
- relations based on a geographical location; and
- websites owned by the same person or company.

From all of these relations I am interested in identifying relations between websites based on webmasters. Webmasters are defined as the people or companies that own a website or are able to edit a website.

By identifying these unknown relationships I provide more structure to the Internet. These relationships are interesting for Internet users as they are able to quickly find all websites from a webmaster. Another important use case is in law enforcement. An example of this is when a webmaster must be identified because of illegal activities on a website. If this webmaster cannot be identified from the illegal website, other websites maintained by the webmaster can be explored. Information may be found on these other websites which can be used to identify the webmaster (e. g. contact information on a personal website).

Relating websites based on ownership is done before by tools, such as SpyOnWeb and DomEye. These kind of tools utilize, among others, identifiers, which are unique across websites, and IP addresses. Additionally, it is suspected that large search engines, such as Google, make use of this information. This research is different as it is able to consider a larger number of identifiers, which are categorized as hard evidence. It is investigated that an identifier shared among multiple websites is an indication for a webmaster relation.

However, the biggest difference, to tools such as SpyOnWeb and DomEye, is that I consider fuzzy evidence as well. Fuzzy evidence is a category which contains features from websites. A feature is any characteristic that can be detected on websites that does not belong to hard evidence. Examples of features are a website's average color, location information and the most used words. The features that I consider in this research consist of website technologies. These technologies include servers, programming languages and frameworks.

Technologies are used as webmasters tend to use the same familiar techniques on multiple websites. Besides being familiar with techniques it is easier to maintain a collection of websites build with the same techniques.

Where identifiers directly indicate relations between websites, features indicate relations when combined. To be able to combine features, a logistic regression algorithm is used, which is a supervised learning algorithm. As input for the algorithm, features are served with identifiers as a base for the ground truth.

Since website relations are investigated, the features that are served as input must represent pairs of websites. Each website is represented by a feature vector, which is a list of booleans. For each feature the boolean indicates whether that feature is detected on a website. In order to combine feature vectors from two websites 6 methods are explored in total:

- sum of shared features;
- concatenation;
- conjunction;
- addition;
- conjunction and disjunction; and
- full factorial.

Each method is implemented and the results are evaluated in Figure 11. The results are measured by a combination of precision and recall, called F_1 score. The F_1 score for the methods are between 0.79 and 0.91, where the full factorial method is the method with the best performance. An F_1 score of 0.91 can be interpreted as being able to correctly predict whether a pair is related or unrelated for 9 out of 10 pairs.

These results are based on a dataset gathered by a team of volunteers from the Common Crawl Foundation. This dataset is created with the aim that researchers do not have to focus on gathering the data, but can instead focus on performing research. Creating a web corpus of high quality is a difficult task. It is difficult to select seeds, the crawler's starting point, that can be used to create a dataset representative for the Internet. It is also difficult to avoid crawling spam and creating a crawler that is not affected by Search Engine Optimization (SEO).

From the Common Crawl dataset, which consists of several billions of websites, only 20.000 websites are considered of which 200 million pairs can be created. This is a small portion of the 1 billion websites that are available, which means that this work must be performed on bigger datasets to increase its reliability.

I answer two research questions by investigating relations that can exist between websites and investigating which characteristics that can be used for identifying relations. Another two questions are answered by combining features for the identification of relations and evaluating the results.

With answers to these four questions, finally an answer can be given to the main research question:

MRQ How can unknown relations between websites be identified?

This question is partly answered by research in to hard evidence. By using identifiers it is possible to identify unknown webmaster relations. Research into fuzzy evidence has proven that combinations of features can also be used to identify these unknown relations.

In conclusion, this work provides a basis for identifying unknown webmaster relations between websites.

APPENDIX

A

HARD EVIDENCE

This Appendix comprises the types of hard evidence implemented in their respective categories. Each category contains a Table summarizing each type of hard evidence. The Tables contain the name of the hard evidence, a reference to its website and a code snippet. Inside the code snippet the identifiers are marked with: ID.

A.1 ANALYTICS

Table 9 lists all identifiers implemented in the analytics category.

Table 9: The types of analytics identifiers implemented, with a reference to the website and code snippets

NAME	WEBSITE	CODE SNIPPET

Alexa	www.alexa.com/	`<script type="text/javascript">`
		`_atrk_opts = { atrk_acct: "ID", domain:"domain.com",dynamic: true};`
		`(function() {`
		`  var as = document.createElement('script'); as.type = 'text/javascript';`
		`  as.async = true;`
		`  as.src = "https://d31qbv1cthcecs.cloudfront.net/atrk.js";`
		`  var s = document.getElementsByTagName('script')[0]; s.parentNode.`
		`  insertBefore(as, s);`
		`})();`
		`</script>`
		`<noscript><img src="https://d5nxst8fruw4z.cloudfront.net/atrk.gif?account=`
		`  ID" style="display:none" height="1" width="1" alt="" /></noscript>`

Chartbeat	www.chartbeat.com/	`_sf_async_config.uid = ID;`
		`or`
		`_sf_async_config={uid:ID};`

ComScore	www.comscore.com/	`<!-- Begin comScore Tag -->`
		`<script>`
		`  var _comscore = _comscore || [];`
		`  _comscore.push({ c1: "2", c2: "[ID]" });`
		`  (function() {`
		`  var s = document.createElement("script"), el =`
		`  document.getElementsByTagName("script")[0]; s.async = true;`
		`  s.src = (document.location.protocol == "https:" ? "https://sb" : "http`
		`  ://b") +`
		`  ".scorecardresearch.com/beacon.js";`
		`  el.parentNode.insertBefore(s, el);`
		`  })();`
		`</script>`
		`<noscript>`
		`  <img src="http://b.scorecardresearch.com/p?c1=2&c2=[ID]&c4=[PAGE URL]"/>`
		`</noscript>`
		`<!-- End comScore Tag -->`

Table 9: (continued)

NAME	WEBSITE	CODE SNIPPET

Google Analytics	www.google.com/analytics/	`<script type="text/javascript">`
		`var _gaq = _gaq || [];`
		`_gaq.push(['_setAccount', 'UA-ID-ID']);`
		`_gaq.push(['_trackPageview']);`
		`(function() {`
		`  var ga = document.createElement('script'); ga.type = 'text/javascript';`
		`  ga.async = true;`
		`  ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http`
		`  ://www') + '.google-analytics.com/ga.js';`
		`  var s = document.getElementsByTagName('script')[0]; s.parentNode.`
		`  insertBefore(ga, s);`
		`})();`
		`</script>`

Marketo	www.marketo.com/	`<!-- marketo -->`
		`<script type="text/javascript">`
		`(function() {`
		`  var didInit = false;`
		`  function initMunchkin() {`
		`  if(didInit === false) {`
		`  didInit = true;`
		`  Munchkin.init('ID-ID-ID', {cookieAnon: true});`
		`  }`
		`  }`
		`})();`
		`</script>`
		`<!-- end marketo -->`

New Relic	www.newrelic.com/	`<script type="text/javascript">`
		`window.NREUM||(NREUM={});NREUM.info={"applicationID":"appID","`
		`  applicationTime":74,"beacon":"bam.nr-data.net","queueTime":0,"`
		`  licenseKey":"ID","transactionName":"transactionName=","agent":"js-`
		`  agent.newrelic.com\/nr-632.min.js","errorBeacon":"bam.nr-data.net"}`
		`</script>`

QuantCast	www.quantcast.com/	`<script type="text/javascript">`
		`_qevents.push({`
		`  qacct:"p-ID",`
		`  labels:"labelgoeshere"`
		`});`
		`</script>`

Yahoo	www.web.analytics.yahoo.com/	`<script type="text/javascript">`
		`var YWATracker = YWA.getTracker("ID");`
		`YWATracker.submit();`
		`</script>`

Yandex	www.yandex.com/	`<!-- Yandex.Metrika counter -->`
		`<script type="text/javascript">`
		`(function (d, w, c) {`
		`  (w[c] = w[c] || []).push(function() {`
		`  try {`
		`  w.yaCounterID = new Ya.Metrika({`
		`  id:ID,`
		`  webvisor:true,`
		`  clickmap:true,`
		`  trackLinks:true,`
		`  accurateTrackBounce:true`
		`  });`
		`  } catch(e) { }`
		`  });`
		`});`
		`</script>`
		`<!-- /Yandex.Metrika counter -->`

A.2 ADVERTISING

The implemented advertising identifiers are described in Table 10.

Table 10: The types of advertising identifiers implemented, with a reference to their website and code snippets

NAME	WEBSITE	CODE SNIPPET
AdRoll[1]	www.adroll.com	`try { adroll_adv_id = "ID1"; adroll_pix_id = "ID2"; __adroll_loaded = !0; var a = document.createElement("script"), c = "https:" == document.location.protocol ? "https://s.adroll.com" : "http://a.adroll.com"; a.setAttribute("async", "true"); a.type = "text/javascript"; a.src = c + "/j/roundtrip.js"; ((document.getElementsByTagName("head") || [null])[0] || document.getElementsByTagName("script")[0].parentNode).appendChild(a) }catch(e) { b.utils.log(e) }`
AdSense	www.adsense.com	`<!-- Begin Google Adsense code --> <script type="text/javascript"> google_ad_client = "ID"; google_ad_slot = "ad-slot-code-goes-here"; google_ad_width = 300; google_ad_height = 250; </script> <script type="text/javascript" src="http://pagead2.googlesyndication.com/pagead/show_ads.js"> </script> <!-- End Google Adsense code -->`
AdWords	www.adwords.com	`<script type="text/javascript">` `/* <![CDATA[ */` `var google_conversion_id = ID;` `var google_conversion_language = "en";` `var google_conversion_format = "3";` `var google_conversion_color = "ffffff";` `var google_conversion_label = "AAAAAAAAAAAAAAAAAA";` `var google_conversion_value = 0;` `/* ]]> */` `</script>` `<script type="text/javascript" src="//www.googleadservices.com/pagead/conversion.js"></script>` `<noscript>` `<img style="display:none;" src="//www.googleadservices.com/pagead/conversion/ID/?label=AAAAAAAAAAAAAAAAAA&guid=ON&script=0" />` `</noscript>` `or` `<noscript>` `<div style="display:inline;">` `<img height="1" width="1" style="border-style:none;" alt="" src="//googleads.g.doubleclick.net/pagead/viewthroughconversion/ID/?value=0&guid=ON&script=0"/>` `</div>` `</noscript>`
Bronto	www.bronto.com	`<script type="text/javascript">` `var bta = new __bta('ID');` `bta.setHost("email.xxx.com");` `</script>`

1 Consists of two identifiers

Table 10: (continued)

NAME	WEBSITE	CODE SNIPPET

DoubleClick	www.doubleclick.com	`<script async="" defer="" src="//survey.g.doubleclick.net/async_survey?site=ID"></script>`
		
		or
		
		`<script type="text/javascript">`
		`var ord = window.ord || Math.floor(Math.random() * 1e16);`
		`document.write('<script type="text/javascript" src="http://ad.doubleclick.net/ID/adj/waldnet.xl;sz=728x90;pos=2;ord=' + ord + '?"><\/script>');`
		`</script>`
		
		or
		
		`<a rel="nofollow" target="_blank" href="http://pubads.g.doubleclick.net/gampad/clk?id=ID&iu=/nr/IDG.domain.com" onclick="moduleTrack('Sponsored Links','resources-sponsored-links component')"></a>`

A.3 BEACON

The implement beacon identifier is listed in Table 11.

Table 11: The type of beacon identifier implemented, with a reference to its website and code snippet

NAME	WEBSITE	CODE SNIPPET

WebTrends	www.webtrends.com/	`window.webtrendsAsyncInit=function(){`
		`var dcs=new Webtrends.dcs().init({`
		`dcsid:"ID"`
		`,domain:"statse.webtrendslive.com"`
		`,timezone:-8`
		`,plugins:{`
		`hm:{src:"//s.webtrends.com/js/webtrends.hm.js",async:true}`
		`}`
		`}).track();`
		`};`
		`(function(){`
		`var s=document.createElement("script"); s.async=true; s.src="/scripts/webtrends.min.js";`
		`var s2=document.getElementsByTagName("script")[0]; s2.parentNode.insertBefore(s,s2);`
		`}());`

A.4 VERIFICATION METHOD

Table 12 gives an overview of the implemented identifiers used in verification methods.

Table 12: The types of verification identifiers implemented, with a reference to their website and code snippets

NAME	WEBSITE	CODE SNIPPET
Alexa	www.alexa.com/siteowners/claim	`<meta name="alexaVerifyID" content="ID" />`
Baidu	www.zhanzhang.baidu.com/	`<meta name="baidu-site-verification" content="ID" />`
Bing Webmaster tools	www.bing.com/toolbox/webmaster	`<meta name="msvalidate.01" content="ID"></meta>`
Google Webmaster Tools	www.google.com/webmasters/verification/	`<meta name="google-site-verification" content="ID"></meta>`
Google Webmaster Tools [2]	www.google.com/webmasters/verification/	`<meta name="verify-v1" content="ID">`
Pinterest	www.help.pinterest.com/en/articles/verify-your-website	`<meta name="p:domain_verify" content="ID"/>`
Yahoo	www.siteexplorer.search.yahoo.com/	`<meta name="y_key" content="ID"></meta>`
Yandex	www.webmaster.yandex.com/	`<meta name="yandex-verification" content="ID" />`

2 This verification method is replaced by Google Site Verification as of 2009, but is still supported

A.5 MISCELLANEOUS

An overview of the implemented identifiers that belong to the miscellaneous category is given in Table 13.

Table 13: The types of miscellaneous identifiers implemented, with a reference to their website and code snippets

NAME	WEBSITE	CODE SNIPPET

Floodlight Tag Manager	www.google.com/doubleclick/	`<script type='text/javascript'>`
		`//<![CDATA[`
		`var axel = Math.random() + "";`
		`var a = axel * 10000000000000;`
		`  document.write('<iframe src="https://ID.fls.doubleclick.net/activityi;`
		`  src=ID;type=type;cat=cat;ord=' + a + '?" width="1" height="1"`
		`  frameborder="0" style="display:none"></iframe>');`
		`//]]>`
		`</script>`
		`<noscript>`
		`<iframe frameborder='0' height='1' src='https://ID.fls.doubleclick.net/`
		`  activityi;src=ID;type=type;cat=cat;ord=1?' style='display:none'`
		`  width='1'></iframe>`
		`</noscript>`

Facebook SDK	www.developers.facebook.com/	`<meta property="fb:app_id" content="ID">`
		
		`or`
		
		`<script>`
		`  window.fbAsyncInit = function() {`
		`  FB.init({`
		`  appId  : 'ID',`
		`  xfbml  : true,`
		`  version  : 'v2.1'`
		`  });`
		`  };`
		
		`  (function(d, s, id){`
		`  var js, fjs = d.getElementsByTagName(s)[0];`
		`  if (d.getElementById(id)) {return;}`
		`  js = d.createElement(s); js.id = id;`
		`  js.src = "//connect.facebook.net/nl_NL/sdk.js";`
		`  fjs.parentNode.insertBefore(js, fjs);`
		`  }(document, 'script', 'facebook-jssdk'));`
		`</script>`

Google Tag Manager	www.google.com/tagmanager/	`<!-- Google Tag Manager -->`
		`<noscript><iframe src="//www.googletagmanager.com/ns.html?id=GTM-ID"`
		`height="0" width="0" style="display:none;visibility:hidden"></iframe></`
		`  noscript>`
		`<script>(function(w,d,s,l,i){w[l]=w[l]||[];w[l].push({'gtm.start':`
		`new Date().getTime(),event:'gtm.js'});var f=d.getElementsByTagName(s)[0],`
		`j=d.createElement(s),dl=l!='dataLayer'?'&l='+l:'';j.async=true;j.src=`
		`'//www.googletagmanager.com/gtm.js?id='+i+dl;f.parentNode.insertBefore(j,f`
		`  );`
		`})(window,document,'script','dataLayer','GTM-ID');</script>`
		`<!-- End Google Tag Manager -->`

ShareThis	www.sharethis.com/	`<script type="text/javascript">`
		`stLight.options({`
		`  publisher:'ID',`
		`  doNotHash:false,`
		`  doNotCopy:true,`
		`  hashAddressBar:true});`
		`</script>`

B

FUZZY EVIDENCE

An overview of the implemented fuzzy evidence features is given in Table 14.

Table 14: The types of fuzzy evidence features, with a reference to their website and a description

NAME	WEBSITE	DESCRIPTION
AddThis	http://www.addthis.com/	Provides social media sharing buttons for websites
AddToAny	https://www.addtoany.com/	Provides social media sharing buttons for websites
Akamai	https://www.akamai.com/	Content Delivery Network
Apache	http://www.apache.org/	Web server
Apache Tomcat	http://tomcat.apache.org/	Open source web server and servlet container
CakePHP	http://cakephp.org/	PHP framework
CentOS	https://www.centos.org/	Operating system
CloudFlare	https://www.cloudflare.com/	Content Delivery Network
Debian	https://www.debian.org/	Operating system
Django	https://www.djangoproject.com/	Python web framework
Drupal	https://www.drupal.org/	Content Management System
EdgeCast	http://www.edgecast.com/	Content Delivery Network

Table 14: (continued)

NAME	WEBSITE	DESCRIPTION
Facebook SDK	https://developers.facebook.com/docs/apis-and-sdks	Software Development Kit for Facebook
FreeBSD	https://www.freebsd.org/	Operating system
Gentoo	https://www.gentoo.org/	Operating system
Google Code Prettify	https://code.google.com/p/google-code-prettify/	JavaScript library
Google Web Toolkit (GWT)	http://www.gwtproject.org/	Tools for JavaScript front-end applications
Internet Information Systems (IIS)	https://www.iis.net/	Web server
Java	https://www.java.com/	Programming language
JavaServer Pages	http://www.oracle.com/technetwork/java/jsp-138432.html	Programming technology
Java Servlet	http://www.oracle.com/technetwork/java/index-jsp-135475.html	Java programming language program
Joomla	http://www.joomla.org/	Content Management System
JQuery	https://jquery.com/	JavaScript library

Table 14: (continued)

NAME	WEBSITE	DESCRIPTION
JQuery UI	https://jqueryui.com/	JavaScript library
Lightbox	http://lokeshdhakar.com/projects/lightbox/	JavaScript library
Litespeed	https://www.litespeedtech.com/	Web server
Microsoft ASP.NET	https://www.asp.net/	Web application framework
Modernizr	http://modernizr.com/	JavaScript library
Mongrel	https://rubygems.org/gems/mongrel	Web server
Nginx	http://nginx.org/	Web server
OpenGSE	https://code.google.com/p/opengse/	Test suite
Optimizely	https://www.optimizely.com/	Optimization technology
Outbrain	http://www.outbrain.com/	Content recommendation platform
OWL Carousel	http://owlgraphic.com/owlcarousel/	JavaScript library
PHP	https://secure.php.net/	Programming language
Python	https://www.python.org/	Programming language

Table 14: (continued)

NAME	WEBSITE	DESCRIPTION
reCAPTCHA	https://www.google.com/recaptcha/	User-dialogue system
Redhat	http://www.redhat.com/	Operating system
Ruby	https://www.ruby-lang.org/	Programming language
ShareThis	http://www.sharethis.com/	Provides social media sharing buttons for websites
Squarespace	http://www.squarespace.com/	Content Management System
Twitter Bootstrap	http://getbootstrap.com/	Front-end framework
Ubuntu	http://www.ubuntu.com/	Operating system
Unix	http://www.unix.org/	Operating system
Varnish	https://www.varnish-cache.org/	Web accelerator
Windows Server	http://www.microsoft.com/en-us/server-cloud/products/windows-server-2012-r2/default.aspx	Web server
W3 Total Cache	https://wordpress.org/plugins/w3-total-cache/	Web accelerator

Table 14: (continued)

NAME	WEBSITE	DESCRIPTION
WooCommerce	http://www.woothemes.com/woocommerce/	eCommerce toolkit for WordPress
WordPress	https://wordpress.com/	Content Management System

C

TRUTH TABLES OF MERGE METHODS

Truth tables are given for the merge methods used in dataset preparation as outlined in Section 5.2.2. Formally, *addition* in Table 17 is not a truth table as the values do not represent booleans, but numeric values. No truth table can be created for *sum of shared features* as the result is a single value.

Table 15: Concatenation

A	B	A	B
1	1	1	1
1	0	1	0
0	1	0	1
0	0	0	0

Table 16: Conjunction

A	B	A∧B
1	1	1
1	0	0
0	1	0
0	0	0

Table 17: Addition

A	B	A+B
1	1	2
1	0	1
0	1	1
0	0	0

Table 18: Conjunction and Disjunction

A	B	A∧B	A∨B
1	1	1	1
1	0	0	1
0	1	0	1
0	0	0	0

Table 19: Full Factorial

A	B	A	B	A∧B
1	1	1	1	1
1	0	1	0	0
0	1	0	1	0
0	0	0	0	0

D

LOGISTIC REGRESSION FOR FULL FACTORIAL

The first and second part of logistic regression coefficients, sorted lexicographically, are shown in Figure 14. These parts encode individual features for two websites. The logistic regression coefficients for the third part are given in Figure 15. This part encodes shared features between the two websites.

Most of the coefficients for the first two parts in Figure 14a and Figure 14b are close to zero or have standard deviations overlapping with zero. This means that for the features corresponding with these coefficients it cannot be said that they contribute to identifying relations between websites. A few features, such as *Joomla* and *Ubuntu*, have a negative contribution. From a comparison of the two plots it can be said that the coefficients for the features are more or less similar.

(a) First part of logistic regression coefficients for *full factorial*

(b) Second part of logistic regression coefficients for *full factorial*

Figure 14: First and second parts of logistic regression coefficients for *full factorial*

In Figure 15 an overview is given of the third part of features made with *full factorial*. More insight in these features is given in Section 6.2.2. It can be said that most of the features contribute positively to the model. There are some features close to zero, such as *Gentoo* and there is one feature leaning towards the negative, namely *JavaServer Pages*.

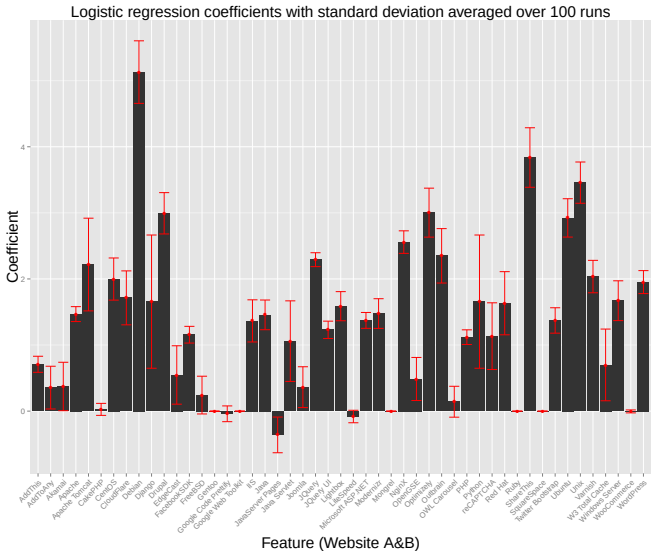


Figure 15: Third part of logistic regression coefficients for *full factorial*

BIBLIOGRAPHY

Ahmed Abbasi and Hsinchun Chen. Applying authorship analysis to extremist-group web forum messages. *Intelligent Systems, IEEE*, 20(5):67–75, 2005. (Cited on page 13.)

Ethem Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2nd edition, 2010. ISBN 026201243X, 9780262012430. (Cited on pages 7, 28, and 36.)

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999. (Cited on page 5.)

Elizabeth Sugar Boese. *Stereotyping the web: genre classification of web documents*. PhD thesis, Citeseer, 2005. (Cited on page 6.)

Danushka T Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Measuring the similarity between implicit semantic relations from the web. In *Proceedings of the 18th international conference on World wide web*, pages 651–660. ACM, 2009. (Cited on page 6.)

Eric A Brewer. Towards robust distributed systems. In *PODC*, volume 7, 2000. (Cited on page 39.)

Sergey Brin. Extracting patterns and relations from the world wide web. In *The World Wide Web and Databases*, pages 172–183. Springer, 1999. (Cited on page 6.)

Declan Butler. Web data predict flu. *Nature News*, 456(7220):287–288, 2008. (Cited on page 7.)

Chia Hui Chang, Mohammed Kayed, Moheb R Girgis, and Khaled F Shaalan. A survey of web information extraction systems. *Knowledge and Data Engineering, IEEE Transactions on*, 18(10):1411–1428, 2006. (Cited on page 5.)

Min Chen, Shiwen Mao, Yin Zhang, and Victor CM Leung. Big data applications. In *Big Data*, pages 59–79. Springer, 2014. (Cited on page 7.)

Gobinda G Chowdhury. Natural language processing. *Annual review of information science and technology*, 37(1):51–89, 2003. (Cited on page 6.)

Kimberly M Christopherson. The positive and negative implications of anonymity in internet social interactions:"on the internet, nobody knows you're a dog". *Computers in Human Behavior*, 23(6): 3038–3056, 2007. (Cited on page 1.)

Jennifer Chu-Carroll, Eric W Brown, Adam Lally, and J William Murdock. Identifying implicit relationships. *IBM Journal of Research and Development*, 56(3.4):12–1, 2012. (Cited on page 6.)

Jim Cowie and Wendy Lehnert. Information extraction. *Commun. ACM*, 39(1):80–91, jan 1996. ISSN 0001-0782. doi: 10.1145/234173.234209. URL http://doi.acm.org/10.1145/234173.234209. (Cited on page 5.)

Kenneth Cukier. *Data, data everywhere: A special report on managing information*. Economist Newspaper, 2010. (Cited on page 7.)

Aron Culotta and Jeffrey Sorensen. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 423. Association for Computational Linguistics, 2004. (Cited on page 6.)

M. Cutler, H. Deng, S.S. Maniccam, and W. Meng. A new study on using html structures to improve retrieval. In *Tools with Artificial Intelligence, 1999. Proceedings. 11th IEEE International Conference on*, pages 406–409, 1999. doi: 10.1109/TAI.1999.809831. (Cited on page 5.)

Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008. (Cited on page 39.)

Muhammad Faheem. Intelligent crawling of web applications for web archiving. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 127–132. ACM, 2012. (Cited on pages 6 and 7.)

Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI magazine*, 17(3):37, 1996. (Cited on page 5.)

Dennis Fetterly, Mark Manasse, and Marc Najork. Spam, damn spam, and statistics: Using statistical analysis to locate spam web pages. In *Proceedings of the 7th International Workshop on the Web and Databases: colocated with ACM SIGMOD/PODS 2004*, pages 1–6. ACM, 2004. (Cited on page 43.)

Common Crawl Foundation. Common crawl, August 2015. URL https://commoncrawl.org/. (Cited on page 43.)

Dayne Freitag and Andrew McCallum. Information extraction with hmms and shrinkage. In *Proceedings of the AAAI-99 workshop on machine learning for information extraction*, pages 31–36. Orlando, Florida, 1999. (Cited on pages 5 and 6.)

Robert Gaizauskas and Yorick Wilks. Information extraction: Beyond document retrieval. *Journal of documentation*, 54(1):70–105, 1998. (Cited on page 5.)

Lars George. *HBase: the definitive guide*. " O'Reilly Media, Inc.", 2011. (Cited on page 39.)

Clinton Gormley and Zachary Tong. *Elasticsearch: The Definitive Guide*. O'Reilly & Associates, 2014. (Cited on page 39.)

Apache Hadoop. Apache hadoop, March 2015. URL http://hadoop.apache.org/. (Cited on pages 38 and 39.)

Robin Hecht and Stefan Jablonski. Nosql evaluation. In *International Conference on Cloud and Service Computing*, pages 336–41, 2011. (Cited on page 7.)

Miguel Helft. Google uses searches to track flu's spread. *The New York Times*, *http://www.nytimes.com/2008/11/12/technology/internet/12flu.html*, 2008. (Cited on page 7.)

Cambridge Intelligence. Keylines website. August 2015. URL http://keylines.com/. (Cited on page 40.)

Adam Jacobs. The pathologies of big data. *Communications of the ACM*, 52(8):36–44, 2009. (Cited on page 7.)

Sun Kim and Byoung-Tak Zhang. Evolutionary learning of web-document structure for information retrieval. In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, volume 2, pages 1253–1260. IEEE, 2001. (Cited on page 5.)

Moshe Koppel, Jonathan Schler, Shlomo Argamon, and Eran Messeri. Authorship attribution with thousands of candidate authors. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 659–660. ACM, 2006. (Cited on page 13.)

Moshe Koppel, Jonathan Schler, and Shlomo Argamon. Computational methods in authorship attribution. *Journal of the American Society for information Science and Technology*, 60(1):9–26, 2009. (Cited on page 13.)

Chakravanti Rajagopalachari Kothari. *Research methodology: methods and techniques*. New Age International, 2011. (Cited on page 3.)

Balachander Krishnamurthy and Craig E Wills. On the leakage of personally identifiable information via online social networks. In *Proceedings of the 2nd ACM workshop on Online social networks*, pages 7–12. ACM, 2009. (Cited on pages 15 and 21.)

Neal Leavitt. Will nosql databases live up to their promise? *Computer*, 43(2):12–14, 2010. (Cited on page 7.)

Rinaldo Lima, Bernard Espinasse, Hilário Oliveira, Laura Pentagrossa, and Fred Freitas. Information extraction from the web: An ontology-based method using inductive logic programming. In *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, pages 741–748. IEEE, 2013. (Cited on page 6.)

Christoph Lindemann and Lars Littig. Coarse-grained classification of web sites by their structural properties. In *Proceedings of the 8th annual ACM international workshop on Web information and data management*, pages 35–42. ACM, 2006. (Cited on page 6.)

Christoph Lindemann and Lars Littig. Classifying web sites. In *Proceedings of the 16th international conference on World Wide Web*, pages 1143–1144. ACM, 2007. (Cited on page 6.)

Apache Lucene. Apache lucene, March 2015. URL `http://lucene.apache.org/core/`. (Cited on page 39.)

David Madigan, Alexander Genkin, David D Lewis, Shlomo Argamon, Dmitriy Fradkin, and Li Ye. Author identification on the large scale. In *Proc. of the Meeting of the Classification Society of North America*, 2005. (Cited on page 13.)

Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008. (Cited on page 5.)

David Martin, Hailin Wu, and Adil Alsaid. Hidden surveillance by web sites: Web bugs in contemporary use. *Communications of the ACM*, 46(12):258–264, 2003. (Cited on page 23.)

Blaz Novak. A survey of focused web crawling algorithms. 2004. (Cited on page 6.)

United States Government Accountability Office. Privacy: Alternatives exist for enhancing protection of personally identifiable information. Technical Report GAO-08-536, United States Government Accountability Office, May 2008. (Cited on page 24.)

Christopher Olston and Marc Najork. Web crawling. *Foundations and Trends in Information Retrieval*, 4(3):175–246, 2010. (Cited on pages 6 and 7.)

Han Woo Park, George A Barnett, and In-Yong Nam. Hyperlink–affiliation network structure of top web sites: Examining affiliates with hyperlink in korea. *Journal of the American Society for Information Science and Technology*, 53(7):592–601, 2002. (Cited on pages 14 and 15.)

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. (Cited on page 40.)

Xiaoguang Qi and Brian D Davison. Web page classification: Features and algorithms. *ACM Computing Surveys (CSUR)*, 41(2):12, 2009. (Cited on page 6.)

Lee Rainie, Sara Kiesler, Ruogu Kang, Mary Madden, Maeve Duggan, Stephanie Brown, and Laura Dabbish. Anonymity, privacy, and security online. *Pew Research Center*, 2013. (Cited on page 1.)

Marko A. Rodriguez, Pavel Yaskevich, Darrick Wiebe, Peter Neubauer, Joshua Shinavier, Ketrina Yim, and Pierre De Wilde. Gremlin website. August 2015. URL https://github.com/tinkerpop/gremlin/. (Cited on page 40.)

Gerard Salton. Automatic text processing: The transformation, analysis, and retrieval of. *Reading: Addison-Wesley*, 1989. (Cited on page 5.)

Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. The hadoop distributed file system. In *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*, pages 1–10. IEEE, 2010. (Cited on page 38.)

Janice C Sipior, Burke T Ward, and Ruben A Mendoza. Online privacy concerns associated with cookies, flash cookies, and web beacons. *Journal of Internet Commerce*, 10(1):1–16, 2011. (Cited on page 23.)

Efstathios Stamatatos. A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, 60(3):538–556, 2009. (Cited on pages 13 and 14.)

Internet Live Stats. Internet live stats. Online, 2015. URL http://www.internetlivestats.com/. (Cited on page 1.)

Michael Stonebraker. Sql databases v. nosql databases. *Communications of the ACM*, 53(4):10–11, 2010. (Cited on page 7.)

John Suler. The online disinhibition effect. *Cyberpsychology & behavior*, 7(3):321–326, 2004. (Cited on page 1.)

Mihai Surdeanu, Sanda Harabagiu, John Williams, and Paul Aarseth. Using predicate-argument structures for information extraction. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 8–15. Association for Computational Linguistics, 2003. (Cited on page 6.)

Thinkaurelius. Titan website. August 2015. URL https://thinkaurelius.github.io/titan/. (Cited on page 40.)

USA TODAY. Stephen hawking opens ups. Online, 2014. URL http://usatoday30.usatoday.com/MONEY/usaedition/2014-12-02-QampA-with-Stephen-Hawking_ST_U.htm. (Cited on page v.)

Daniel Waisberg and Avinash Kaushik. Web analytics 2.0: empowering customer centricity. *The original Search Engine Marketing Journal*, 2(1):5–11, 2009. (Cited on page 22.)

Tom White. *Hadoop: the definitive guide*. " O'Reilly Media, Inc.", 2009. (Cited on pages 38 and 39.)

Hwanjo Yu, Jiawei Han, and KC-C Chang. Pebl: Web page classification without negative examples. *Knowledge and Data Engineering, IEEE Transactions on*, 16(1):70–81, 2004. (Cited on page 6.)

Lotfi A Zadeh. Fuzzy sets. *Information and control*, 8(3):338–353, 1965. (Cited on page 24.)

Sven Meyer Zu Eissen and Benno Stein. Genre classification of web pages. In *KI 2004: Advances in artificial intelligence*, pages 256–269. Springer, 2004. (Cited on page 6.)